TRANSPORTABLE MAPS SOFTWARE

Control Data Corporation

A. E. LaBonte

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441

82 11 02 049
This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-82-194 has been reviewed and is approved for publication.

APPROVED:

JOHN T. BOLAND
Project Engineer

APPROVED:

A. J. DRISCOLL, Colonel, USAF
Chief, Intelligence & Reconnaissance Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRRA) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.
This report covers the development of a transportable software package in FORTRAN for the MAPS image compression technique. Also included are a user's manual for the developed software, a software maintenance manual, and the software listing. Possession of the report will allow a user to install and operate this software on DEC PDP 11/45, 11/70 and VAX 11/780 processors.
PREFACE

This is Volume I of the Final Technical Report on Transportable MAPS Software. It describes the development of the TransMAPS software package; its companion volumes contain the TransMAPS User's Manual (Volume II) and TransMAPS Maintenance Manual (Volume III). This volume is submitted in fulfillment of CDRL item A002 of Contract # F30602-80-C-0326.
# TABLE OF CONTENTS

## VOLUME I. TRANSPORTABLE MAPS SOFTWARE

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>INTRODUCTION AND SUMMARY</td>
<td>I-1</td>
</tr>
<tr>
<td>1.1</td>
<td>Background and Objectives</td>
<td>I-1</td>
</tr>
<tr>
<td>1.2</td>
<td>Software Development</td>
<td>I-2</td>
</tr>
<tr>
<td>1.3</td>
<td>The TransMAPS Package</td>
<td>I-3</td>
</tr>
<tr>
<td>1.4</td>
<td>Documentation Organization</td>
<td>I-5</td>
</tr>
<tr>
<td>2.0</td>
<td>USER OPTION ASSESSMENT</td>
<td>I-8</td>
</tr>
<tr>
<td>2.1</td>
<td>Automated Selections</td>
<td>I-8</td>
</tr>
<tr>
<td>2.2</td>
<td>Option Groupings</td>
<td>I-12</td>
</tr>
<tr>
<td>2.3</td>
<td>Compatibility Constraints</td>
<td>I-13</td>
</tr>
<tr>
<td>3.0</td>
<td>PROCESS PARTITION</td>
<td>I-18</td>
</tr>
<tr>
<td>3.1</td>
<td>Task Space Limitations</td>
<td>I-18</td>
</tr>
<tr>
<td>3.2</td>
<td>Array Address Limitations</td>
<td>I-19</td>
</tr>
<tr>
<td>3.3</td>
<td>System Overhead</td>
<td>I-20</td>
</tr>
<tr>
<td>3.4</td>
<td>Intermodule File Communication</td>
<td>I-21</td>
</tr>
<tr>
<td>4.0</td>
<td>IMPLEMENTATION</td>
<td>I-24</td>
</tr>
<tr>
<td>4.1</td>
<td>Transportability</td>
<td>I-24</td>
</tr>
<tr>
<td>4.2</td>
<td>Modularization</td>
<td>I-25</td>
</tr>
<tr>
<td>4.3</td>
<td>Raster Image File Formats</td>
<td>I-26</td>
</tr>
<tr>
<td>4.4</td>
<td>Subframe Image File Formats</td>
<td>I-27</td>
</tr>
<tr>
<td>4.5</td>
<td>Comment Guidelines</td>
<td>I-28</td>
</tr>
<tr>
<td>5.0</td>
<td>TESTING</td>
<td>I-30</td>
</tr>
<tr>
<td>5.1</td>
<td>Compression-Logic Diagnostic Test Image</td>
<td>I-30</td>
</tr>
<tr>
<td>5.2</td>
<td>Product-Generation Familiarization Test Image</td>
<td>I-33</td>
</tr>
<tr>
<td>5.3</td>
<td>Annotation Option Test Patterns</td>
<td>I-33</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Example of TransMAPS Software Application</td>
<td>6</td>
</tr>
<tr>
<td>2-1</td>
<td>MAPS Compression Planning Form</td>
<td>14</td>
</tr>
<tr>
<td>2-2</td>
<td>MAPS Product Planning Form</td>
<td>15</td>
</tr>
<tr>
<td>2-3</td>
<td>MAPS Image Assembly and Annotation Planning Form</td>
<td>16</td>
</tr>
<tr>
<td>3-1</td>
<td>Overall TransMAPS Structure</td>
<td>23</td>
</tr>
<tr>
<td>5-1</td>
<td>MAPS Compression-Logic Diagnostic Image</td>
<td>31</td>
</tr>
<tr>
<td>5-2</td>
<td>MAPS Familiarization Test Image</td>
<td>34</td>
</tr>
<tr>
<td>5-3</td>
<td>Annotation Options Test Pattern</td>
<td>35</td>
</tr>
</tbody>
</table>
TRANSPORTABLE MAPS SOFTWARE

SECTION ONE

INTRODUCTION AND SUMMARY

Micro-Adaptive Picture Sequencing (MAPS) is a computationally-efficient, contrast-adaptive, variable-resolution spatial image encoding technique. This document describes the implementation of the general MAPS process as a transportable software ensemble which will hereafter be referred to as the TransMAPS Package or simply as 'TransMAPS'.

1.1 Background and Objectives

Micro-Adaptive Picture Sequencing originated in the Information Sciences Division of Control Data Corporation and has undergone extensive further development and exploration with sponsorship from the Rome Air Development Center. These efforts are summarized in the following reports and articles:

References:

RADC MAPS-Related Reports:


SPIE Proceedings Articles:


These prior investigations have dealt with many different facets of the image representation problem including image partitioning, large-area or macro-fidelity control, local or micro-fidelity control, coding strategies, and artifact-masking or perceptual quality enhancement in the reconstructed imagery. Through these studies, MAPS has evolved into a mature set of processes which still retains a rich space of user options. The present effort integrates these previous developments to form the TransMAPS software package.

The general objectives of this implementation are to coordinate existing MAPS results and to make MAPS functionally available to a much wider community of potential users. TransMAPS fulfills these objectives through use of a readily-available language (FORTRAN) running on widely-used computer systems (DEC PDP-11 and VAX). More specific objectives emphasize transportability within this DEC system environment and interactive capabilities to support a broad spectrum of user experience levels and applications intent.

1.2 Software Development

Four main activities were involved in the TransMAPS development cycle. First was evaluation and coordination of the large space of MAPS user options which arose as the technique evolved. Here the principal problems involved assessment of potential conflicts and automation of those features where a clear-cut choice could be made based on the results of prior investigations. Note that overall organization of TransMAPS is determined primarily by the structure which results from this option-space evaluation. Moreover, at least one and sometimes several detailed 'algorithms' are already available for each of the subprocesses invoked by each selected option. Thus, the option assessment really becomes the first stage in the top-down TransMAPS software design.
Again, because of the algorithmic detail already known, the second stage of the development cycle focused on the constraints and compatibility issues raised by the characteristics of the host computer. Here, the sixteen-bit word length of the PDP-11 systems was the dominant controlling limitation. The resulting address field constraints together with the option structure imposed the requirements for extensive partitioning of the MAPS process.

Implementation considerations formed the core of the third major activity group. Issues of modularity, transportable constructs, file environment, and internal program documentation guidelines were principal subtopics. This stage carried development through detailed TransMAPS code preparation.

The fourth phase involved comprehensive program testing followed by TransMAPS installation and demonstration on the target PDP-11/70 in the RADC Image Processing System.

1.3 The TransMAPS Package

The fact that TransMAPS is an integrated software 'package' and not just a collection of computer programs is implicit in all of the development efforts. This package contains seven interrelated main program modules, six data sets, a MAPS standard file structure, a set of pre-planning aids for user interaction support, and the TransMAPS user and maintenance documentation. The contents of these five categories is refined one more level in the following tabulation:
TransMAPS Package:

Seven Main Program Modules:

#1 SUBFR M Raster to Subframe Conversion - SF.FSV, SF.OBJ
#2 MAPS MAPS Compression - MP.FSV, MP.OBJ
#3 DMAPS MAPS Decompression & Level Image Formation - DM.FSV, DM.OBJ
#4 ADAPT MAPS Adaptive Image Smoothing - AD.FSV, AD.OBJ
#5 DIFFER MAPS Difference Image Formation - DF.FSV, DF.OBJ
#6 RASTP Subframe to Raster Conversion - RS.FSV, RS.OBJ
#7 ANNUTE Image Assembly and Annotation - AI.FSV, AI.OBJ

Provided on Computer-Compatible Tape (CCT):
FORTAN IV-Plus Source Code = x.FSV
FORTAN Object (F4P) Code = x.OBJ

Six Data Sets:

--- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- -----
The first six program modules all implement processes directly related to MAPS. This set of six further divides into modules #1 and #2 which deal with compression and modules #3 through #6 which provide the generation of MAPS output 'products'. Four product types are available: MAPS directly-decompressed tonal images, MAPS resolution or level images, MAPS adaptively-smoothed tonal images, and MAPS difference images. The 'level' images give direct display of the pattern of variable resolution generated by the MAPS compression process. The 'difference' images display the fidelity between source and product images in terms of either a 'signed' error with a neutral gray zero-point bias or an 'amplified' absolute error.

The seventh module provides a stand-alone image assembly and annotation capability. It is included to support the display of the MAPS products but can be used to format and label imagery from other sources as well. Figure 1-1 exhibits a source image, its MAPS tonal decompression, the corresponding resolution image, and an amplified difference image assembled and annotated using module #7. The original image in this example is 480 lines x 624 pixels x 8 bits; the MAPS compression level is 2 bits/original pixel; and the difference amplification factor is 10. In the resolution image, successively lighter regions correspond to finer levels of local MAPS resolution. The difference image has been complemented (during the assembly process) so the lighter flecks represent larger absolute source-to-MAPS differences. The sample in Figure 1-1 is intended to give the flavor of the capabilities of the TransMAPS package and to illustrate them with 'real-world' imagery (in this case a frame of 'video' size).

1.4 Documentation Organization

The complete documentation for TransMAPS is distributed across three volumes: this final report, the TransMAPS User's Manual, and the TransMAPS Maintenance Manual. The User's Manual emphasizes the basis of
Figure 1-1. Example of TransMAPS Software Application
the MAPS subprocesses, the space of user options, and the interactive protocols needed to apply the package as an image coding system. The focus in the User's Manual is thus on MAPS as an integrated family of techniques and on the concepts which underlie them. The user may either apply MAPS for functional image compression or may use TransMAPS to become familiar with and explore the MAPS conceptual and process structure. The perspective here is on 'what it does'.

The Maintenance Manual emphasizes the system and specific implementation aspects of TransMAPS. It addresses MAPS as hosted by the computer system and focuses on installation, modification, and code maintenance issues. The Maintenance Manual contains complete COMMENT-annotated source listings of the seven program modules and these listings form the ultimate level of detailed implementation documentation. The perspective here is on 'how it's done'.

The present document - the project final report - is presumed to contain both the User's Manual and the Maintenance Manual in its scope. They are simply published as separate documents for the convenience of later use where their roles are quite different. The remaining topics which must be covered in this document, then, involve descriptions of the activities specific to the original construction of TransMAPS; in essence, the 'scaffolding' of the TransMAPS development. Four topics - user option assessment, process partition, implementation, and testing - are discussed briefly in the sections which complete this volume. Readers unfamiliar with the basic MAPS constructs may wish to read the synopsis of these processes in Section Six of the User's Manual to gain a framework for the rest of the documents. Alternatively, the references outlined in Section 1.1 may be consulted for this purpose.
SECTION TWO

USER OPTION ASSESSMENT

The organization of TransMAPS is determined primarily by the structure imposed on the space of user options. Throughout previous development and refinements of the MAPS technique, many alternate configurations and strategies were investigated. This resulted in a very large space of potential user options. Thus, the first task in the TransMAPS integration involved three subtasks focused on this option space: reduction of the space by 'automating' some of the selections; sequencing and 'grouping' the retained options to structure the process; and assessing the 'compatibility constraints' within this structure. Each of these subtasks will be dealt with in turn.

2.1 Automated Selections

The variety and nuances of possible MAPS alternatives is bewildering. Thus, any reduction in this option space which does not significantly degrade flexibility or performance will enhance the usability of TransMAPS. Two types of option selection 'automation' were sought. In the first type, the selection is made implicit as a part of another user action so that it becomes transparent. In the second type, the range of states for a particular option dimension is restricted to those where clear advantages have been demonstrated.

Three subclasses of implicit selections arose - in image staging, in process mode, and in generic options. In image staging, the DEC systems are heavily disk-file oriented and do not support simple direct magnetic tape interaction under FORTRAN. Thus, file skipping and positioning occur only on transfer of image files from tape to disk and not on every processing pass through a given source file. Moreover, communication of
image data between processes is disk-based and this communication can be made transparent by implementation of a MAPS file structure with standard naming conventions. The staging tasks are then handled either automatically or as part of the initial data entry and need not be made explicit.

Specification of the MAPS process mode involves selection of the process(es) to be run on the source imagery and their sequence. These choices are made implicitly as the user invokes the corresponding TransMAPS program modules. Furthermore, since intermodule communication is accomplished through the standard file structure, the user gains flexibility in sequence specification because the required intermediate results are generated and saved automatically. Thus, the full mode selection need not be developed beforehand - invoking TransMAPS modules remains interactive.

Generic options allow more specific option types to be absorbed as special cases and thus simplify the explicit tree of options without reducing flexibility. One potential option absorbed in this manner is the 2x2-element-sharpening heuristic from the adaptive smoothing module which is superseded by the more powerful subelement pattern coding which applies to 2x2 and larger MAPS elements if desired. Even more generality is represented by the use of piecewise-linear code-space-to-contrast-space and code-space-to-intensity-space remapping specifications. More specific options such as log or exponential remappings can be approximated by and thus absorbed into the process of defining suitable breakpoints for the piecewise-linear mappings. Specification of the required intensity-space-to-code-space demapping is automated by inverting the code-to-intensity remap.

Five subclasses of state-restriction selections arose - in file annotation, in performance evaluation, in resolution image generation, in pattern bias specification, and in adaptive smoothing control. Since the
MAPS file structure must serve intermodule communication, certain ancillary information such as image size and partition parameters must be included as part of the file information. Thus, some form of file header is required and a standard header format was chosen. This format includes an 'image name' field and is added automatically to all of the standard MAPS files. Thus, the user no longer need choose whether to provide such annotation or not.

Performance evaluation for both compression level and fidelity in overall image terms is a simple by-product of the statistics which must be gathered for determining optimal pattern biases. Thus, these evaluations have a negligible impact and are automated as part of the compression process. Gathering data for more detailed fidelity distributions, however, does have a significant impact on compression computational efficiency. Moreover, if adaptive smoothing is to be applied, the process is sufficiently non-linear so that difference statistics can only be gathered during or after the smoothing process. Thus, more detailed fidelity performance analysis is deferred until the point where MAPS difference images are formed. These statistics are easily obtained there so they are included as an automatic rather than optional by-product in the difference process.

The MAPS resolution or level image is a useful evaluation tool to assess how MAPS is distributing its resources relative to the image content. The information in this resolution image is also a necessary input to the MAPS adaptive smoothing process. Thus, formation of the level image is conveniently automated as part of the direct MAPS tonal image decompression. The only remaining potential option in this case, then, is the selection of the gray-scale values to be associated with each level. This association must present the 'level' image itself in a discernible form and must make the information easily accessible for
adaptive smoothing. A suitable selection which automates the gray-scale assignment and meets these requirements is one in which the MAPS level or resolution codes are simply stored in the upper three bits of each level image byte.

Selection of appropriate pattern biases to be used with the MAPS subelement pattern coding mode can be optimized in a mean-square-error sense and automated based on statistics gathered during the compression step. Thus, the imagery itself yields the best choice for these bias values and the user need not be burdened with their separate specification.

In MAPS adaptive smoothing, the 'adaptation' includes three different considerations: the size of the 'convolution' window, activation of surrounding elements based on their size relative to the target element, and activation of surrounding elements based on their intensity relative to the target element. Although specification of each of these could be left to the user's control, specific choices perform well and have significant a priori justification. Thus, the window size is chosen to have an edge which is one cell smaller than the target element. This makes the window 'odd' (symmetric about its center pixel). This choice is consistent with the assertion that the target element size reflects the local intensity correlation length in the image. The 'surround' activation based on size is chosen to include all elements no smaller than one resolution level below the target element. This is consistent with the assertion that finer relative resolution is a priori evidence of localized image activity and should not be included in the 'convolution'. Finally, the selection of weighting functions for the convolution is restricted to a choice between uniform weight over the window or two-dimensional Gaussian weight with user specified spread. Uniform weighting has a particularly simple implementation and Gaussian weighting exhibited slight superiority among the various functions investigated in previous MAPS studies.
2.2 Option Groupings

Following the pruning of the option tree, the next major task in the user option assessment is to order the remaining options and to cluster them into groups according to the processes which require them. The ordering chosen is essentially the sequence in which the option selections are used. The process groupings are as follows:

Option Groupings:

Raster to Subframe Conversion:

Source Image Identification
Source Image Position Specification
Source Image Size Specification
Source Image Partition Specification

MAPS Compression:

User Mode Selection
Macro-Fidelity Control
Micro-Fidelity Control
Gray-Scale Manipulations

MAPS Decompression and Resolution Image Formation:

(User Transparent)

MAPS Adaptive Image Smoothing:

Convolution weighting Specification
Ditmer Amplitude Specification

MAPS Difference Image Formation:

Input Image Pair Selection
Difference Image Control

Subframe to Master Conversion:

Output Product Image Type Selection

Image Assembly and Annotation:

Output Image Specification
Embedded Input Image Specifications
Embedded Annotation Specifications

Note that this tabulation displays only the first level of refinement under each process. Particularly in the MAPS Compression and the Image Assembly and Annotation processes, several further levels are needed.
This option hierarchy coupled with an appropriate default structure allows the casual or inexperienced user to treat TransMAPS as a 'black box' (or more nearly dark gray) image coding system. Very few parameters need to be specified for simple uniform fidelity coding and reconstruction of the source imagery. On the other hand, the more experienced user or one who seeks to gain more insight into the MAPS processes can selectively penetrate the option tree and tailor the control as desired.

In order to assist with such exploitation or exploration, the option space has been laid out in three detailed 'User Planning Forms'. They are intended to serve as an aid for pre-planning of complex interactive sessions, to provide a convenient record of MAPS processing, and to form a 'road-map' which lays out the entire option space at one time. Reduced versions of these Planning Forms are exhibited here as Figures 2-1 (MAPS Compression), 2-2 (MAPS Product Generation), and 2-3 (Annotation and Image Assembly).

The information on these three forms, then, characterizes the option structure and thus the organization of the interactive TransMAPS package. Further descriptions of the individual option entries is given in the User's Manual, Section Seven.

2.3 Compatibility Constraints

The remaining option analysis effort was directed at uncovering combinations of active options which are internally incompatible. Only two significant problems were found and neither restricts the performance of the package appreciably. Both problems involve the 'staggered' subframe image partition. Both also arise partially due to 'computational considerations' so they are not strict option incompatibilities.
Figure 2-1. MAPS Compression Planning Form
MAPS PLANNING FORM  
(Product Generation)

<table>
<thead>
<tr>
<th>Module</th>
<th>User Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run DMAPS</td>
<td></td>
</tr>
</tbody>
</table>
Files:  
* IN - MAPS.DAT  
* OUT - DMAPS.DAT  
* OUT - LEVEL.DAT |
| Run ADAPT | (Not with stagger grid)  
Files:  
* IN - DMAPS.DAT  
* IN - LEVEL.DAT  
* OUT - ADAPT.DAT |
| Run DIFFER |  
Files:  
* IN - IMAGE.DAT  
* IN - DMAPS.DAT  
* IN - ADAPT.DAT  
* OUT - ERROR.DAT  
* OUT - EPRINT.DAT (listing) |
| Run RASTER |  
Files:  
* IN - IMAGE.DAT  
* IN - DMAPS.DAT  
* IN - LEVEL.DAT  
* IN - ADAPT.DAT  
* IN - ERROR.DAT  
* OUT - IРАST.DAT  
* OUT - DRAST.DAT  
* OUT - LRAST.DAT  
* OUT - ARAST.DAT  
* OUT - ERAST.DAT |

Figure 2-2. MAPS Product Planning Form
### OUTPUT IMAGE:

- **Type:** (MASTER FILE, PRINTOUT LISTING) [x]
- **Lines:**
- **Pixels:**
- **Complement:** (V H) [x]

### EMBEDDED IMAGES:

<table>
<thead>
<tr>
<th>IMAGE</th>
<th>FIDEN</th>
<th>SKIP LINES</th>
<th>SKIP PIXELS</th>
<th>LINES</th>
<th>PIXELS</th>
<th>START LINE</th>
<th>START PIXEL</th>
<th>COMPLEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(V H)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(V H)</td>
</tr>
</tbody>
</table>

### EMBEDDED MESSAGES:

<table>
<thead>
<tr>
<th>ORIG</th>
<th>ORIENT</th>
<th>CHAN</th>
<th>SYMBOL</th>
<th>CENTER</th>
<th>SIZE</th>
<th>LINE</th>
<th>PIXEL</th>
<th>COMP</th>
<th>TEXT</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
<th>40</th>
<th>45</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The first difficulty is among the staggered subframe partition, the larger subframe sizes (16x16 and 32x32), and the maximum array size addressable on a sixteen-bit computer. The essence of the problem is that a 16 or 32 line recirculating buffer is too restrictive on the source image size, and implementation in terms of an 8-line recirculating buffer with successive partial subframe extraction is too disk I/O intensive to achieve a reasonable efficiency level. However, the advantages of a staggered subframe grid in reducing the perceptible 'blockiness' in the reconstructed image are mostly lost in the larger subframes anyway since 16x16 and 32x32 block sizes can usually be discerned independent of the characteristics of their surroundings. Thus, restriction to 8x8 subframes for the staggered grid partition is not a serious operational limitation.

The other difficulty is among the staggered subframe partition, the adaptive smoothing process, and the 'fast' algorithm for the required 'convolution'. This fast algorithm gains a major portion of its efficiency from the fact that the active elements surrounding the target element have invariant sampling point positions given the size (resolution level) of the target. In the staggered mode, the 'surround' for elements along the subframe edges is different from the surround for an element interior to the subframe. Thus, this crucial invariance is lost and a much less efficient sampling strategy would be required. Thus, adaptive smoothing is restricted to square grid subframe partitions. Note, however, that subframe stagger and adaptive smoothing are really competing processes which both seek to decrease the perception of 'blockiness' in the reconstructed MAPS imagery. Stagger has very low computational cost but is only partially effective in hiding 'blockiness'. Adaptive smoothing is very effective in perceptual improvement but also incurs relatively higher computational expense. Adaptive smoothing combined with stagger seems unlikely to give significantly better perceptual quality than is achieved by adaptive smoothing on square grids. Again, no appreciable operational capability appears to be lost through this constraint.
SECTION THREE

PROCESS PARTITION

The sixteen-bit word size of the PDP-11 systems introduces three types of constraint on the TransMAPS development. The three types are: task space limitations, array address limitations, and system overhead costs. Together, these constraints dictate the need to partition MAPS into process modules which can be accommodated individually within the PDP-11 operating environment. The next three subsections discuss the implications of the limitations; the final subsection then describes the partition and the associated intermodule communication structure.

3.1 Task Space Limitations

Even though a typical large PDP-11 system may have several hundred thousand bytes of main memory, individual tasks are limited to 32K if standard FORTRAN constructs are employed. In 'mapped' PDP-11 systems, this restriction can be circumvented by the process of 'windowing' in 4K blocks (with some overhead for window maintenance). In general, such windowing is not directly accessible via FORTRAN except for data space extension through the VIRTUAL array declaration. Although VIRTUAL allows data access outside of the normal 32K task space, it does so at some expense in program efficiency. Thus, the VIRTUAL construct should be exploited only where absolutely needed.

Another strategy for handling programs which exceed the 32K task limit is through the use of overlay structures. However, this approach is appropriate where the excess size is due to executable code which can be overwritten when its functions have been completed. In the TransMAPS process (as with many 'image processing' tasks), the size requirements are dictated more by the image data needs than by the program statements to manipulate that data. Furthermore, many of the subprocesses are
table-driven with table space dominating the size of the task. Again, this is more data-like than program-like so overlays are of marginal utility.

Four guidelines emerged from these considerations:

* Restrict code to FORTRAN constructs, if possible, to simplify transportability and program maintenance;
* Attempt explicit process partition into modules which reflect the logical structure of MAPS and avoid the complexity inherent in overlays;
* Minimize use of the non-standard and relatively slow VIRTUAL declaration wherever possible; and
* Exploit process symmetries to reduce table size for table-driven processes.

3.2 Array Address Limitations

The sixteen-bit word size also limits the maximum number of addressable elements in an array to 32K. Elements could be single-byte, two-byte, four-byte, or eight-byte. However, the basic source image pixel size is single-byte (8 bit or 6 bits right-justified in 8). Thus, image manipulations are much easier to understand if the arrays are directly organized into bytes. But this leads to a problem if 32x32 pixel subframes are to be used since 32 lines of data must be accessed to obtain each subframe. Straightforward implementation of a 32-line buffer would then limit the image edge (line length) to a thousand pixels. This is clearly too restrictive for a general MAPS capability.

Fortunately, both PDP-11 file systems - FCS-11 and RMS-11 - support a DIRECT access capability for binary files of FIXED record length. This makes it possible to develop input and output modules which reorganize the raster source imagery to subframe format by extracting segments and updating subframes from successive swathes of lines. An eight-line swath at four thousand pixels per line will still fit within the 32K array address constraint. This sets an acceptable limit (4000 pixels) on the size of the source image.
Once in subframe format, all subsequent TransMAPS modules can then deal with the data in terms of deterministic record (subframe) size. These sizes are 64 bytes (8x8 subframe), 256 bytes (16x16 subframe), and 1024 bytes (32x32 subframe). Even the largest of these is a small fraction of the available 32K task space. Moreover, all three sizes are sub- or supra-multiples of the basic disk sector length of 512 bytes. The DIRECT mode of disk access exploits this in finding the track and sector for a particular subframe index; a significant efficiency advantage.

3.3 System Overhead

The 32K task space must also accommodate some system overhead in the form of data communication buffers. The space set aside for this purpose is determined by two Task Build parameters - ACTFIL and MAXBUF. ACTFIL determines the maximum number of files which can be active (open) at one time. MAXBUF determines the maximum record size which can be handled on any file. The buffer space is determined by the product of these two. Thus, it is important to restrict the number of files to the minimum needed and to open, read, and close any initial table-loading files before opening image handling files in each module.

Furthermore, where possible, it is desirable to employ file types with a minimum physical RECORDSIZE. For modules employing subframe data in the DIRECT access mode, this size is at least 1024 bytes. For raster organized image files, the record size will depend on the RECORDTYPE.

Another Task Build parameter, UNITS, also affects the system overhead charged against the task space. UNITS must be set at least as large as the largest logical unit number employed in the program. File table space is set aside for each possible logical unit number up to UNITS whether each number is used or not. Thus, use of the small numbers, 1 through 4, for the data files in each module is advantageous. Unit 5 is the default terminal designation and Unit 6 is the default printer value; retention of these seems reasonable for standardization. Extension
beyond the default, UNITS=6, does not appear necessary.

3.4 Intermodule File Communication

The logical partition of the MAPS processes is already supported by the user option evaluation results. MAPS compression, MAPS product formation, and Image Assembly and Annotation form the three major process categories. Compression is further divided into the macro step of subframe reorganization and the micro step of MAPS coding within each subframe. Product formation is also further divided into MAPS decompression with resolution image formation; adaptive smoothing; difference image formation; and reformatting back to raster organization. In summary, a suitable logical partition into seven modules (including process mnemonics) is:

#1 SUBFRM Source raster to subframe conversion;
#2 MAPS MAPS subframe compression;
#3 DMAPS MAPS decompression/level image formation;
#4 ADAPT MAPS adaptive smoothing;
#5 DIFFER MAPS difference image formation;
#6 RASTER Subframe to raster conversion; and
#7 ANNOTE Annotation and image assembly.

The user interaction with each of these modules was outlined in Sectio. Two. However, appropriate image data must also interact with each of these modules and this communication has several requirements. The process should be transparent to the user. This implies the need for standard (dedicated) file names which can be opened automatically when a particular process is invoked. It also means that necessary control parameters from previous processes be carried internal to the file; a standard header format meets this latter need. Moreover, the file names should be mnemonic for the file type but sufficiently unique to avoid confusion with other files likely to reside in the system. Such a combination of dedicated mnemonic file names enables effective and efficient file maintenance activities to be carried out on the TransMAPS data environment even if they are generated over an extended period of time.
A standard configuration of twelve file types was developed for this intermodule communication function. They are exhibited along with the corresponding modules in Figure 3-1. This figure presents the complete macro-structure of TransMAPS; it is the key organizing chart for the entire software package.
Figure 3-1. Overall TransMAPS Structure
SECTION FOUR

IMPLEMENTATION

Detailed program module development is the next broad step in the top-down design which implements the structure of TransMAPS given in Figure 3-1. Here, 'working models' for many of the process steps already existed in the various MAPS software elements available from prior internal Control Data implementations. Thus, much of the effort consisted of coordination and integration, and the TransMAPS implementation phase could emphasize the issues of transportability and its close relative, software maintainability. Both maintenance and transport are aided by use of standard constructs, modularization of code, standardization of file structures, and consistent internal program documentation practices. Each of these areas is discussed briefly in the remainder of this section.

4.1 Transportability

Some of the transportability considerations have already been mentioned; an example is the use of file types supported by both FCS-11 and RMS-11 file systems. More generally, coding was restricted to FORTRAN constructs supported in both DEC compilers - FORTRAN and FORTRAN IV-PLUS - insofar as possible. This restriction was imposed even where small penalties in execution efficiency were known to accrue.

In two instances, TransMAPS employs constructs which only FORTRAN IV-PLUS supports. In ANNOTE, the image assembly and annotation module, extensive use is made of the library shift function, IISHFT. The shift allows very rapid manipulation of the annotation character bit maps in the various resampling operations; any other approach seemed unnecessarily obscure and cumbersome.

The other type of construct limited to FORTRAN IV-PLUS involves use of
Integer*4 (i.e. four byte integer) arithmetic in several calculations needed for performance evaluation and element or subframe counts. Simple FORTRAN supports the Integer*4 data type but only allows Integer*2 arithmetic operations. Although the four-byte arithmetic can be partitioned into two-byte arithmetic with extensive overflow checking, the efficiency penalties seem far in excess of the slight gains in transportability. In any event, such two-byte for four-byte replacement can be treated as a maintenance function in the rare instances where it is required by the system.

4.2 Modularization

To support maintenance activities and possible future software modification or extension, the TransMAPS detailed design is extensively modularized. Including the main routines, the seven principal modules contain a total of sixty-eight routines. The average routine length without COMMENTS but including continuations is less than fifty lines. Although most routines fit on a single page (at least before COMMENTing), the code for the user interactions was not so restricted. For each module, the interactive protocols were grouped in one (or two) routines for ease in location rather than being broken up arbitrarily to achieve artificially short code segments. Moreover, the FORMAT statements are placed at the point of use rather than being collected at the beginning or end of the routine. This allows the interaction to be followed sequentially through its steps by anyone reading the code and should contribute significantly to rapid understanding of its flow.

Each interaction is in the form of a 'TYPE' statement (with FORMAT) which queries the user and gives current default values and allowed ranges where appropriate. This prompt is followed immediately by the corresponding 'ACCEPT' statement which processes the user's response. If applicable, validity checking is included following the response; such data checks are typically set off by indenting the corresponding code in the user interaction routines. Thus, the interaction proceeds in
consistent chunks with the FORMATS providing integrated internal documentation. Except to set off major groups of user parameters, further COMMENTs in these routines were thought to be more intrusive than helpful and were not used.

All routines were restricted to a single entry and a single exit point. Within the routines, flow is either simple sequential, simple conditional (the equivalent of 'if-then-else'), simple iteration ('DO-loop'), or the non-linear recursion implied by the MAPS resolution adaptation. Thus, except for the MAPS recursive sequencing, all control constructs are of the classical 'structured' variety. The MAPS sequencing is also simple and clear from context.

In many instances, iterative control involves several levels of nesting. For these cases, the hierarchy is set off by extensive and consistent use of DO-loop indentation in all TransMAPS routines other than those for user interaction as already described.

4.3 Raster Image File Formats

As noted earlier, DEC FORTRAN does not support direct magnetic tape operations in any simple manner. Thus, the communication of image data to and from the TransMAPS package is via disk file. Since there does not appear to be any generally-accepted standard format for raster image files, a specific file structure was chosen to satisfy other conditions in TransMAPS. In particular, TransMAPS expects input raster images to be in the form of sequential binary files with a SEGMENTED record type. The advantage of this form is that large logical records (up to four thousand bytes) can be accommodated without the necessity to handle correspondingly large physical records. Thus, the buffer space does not have to be allocated on the basis of this largest logical record size and MAXBUF need not be set larger than 1024, the largest expected record of FIXED type. This limit on the buffer overhead, in turn, has allowed TransMAPS to be implemented without resorting to the VIRTUAL declaration
in any of the modules; a significant gain in efficiency! Even with this restriction, three of the seven modules are well over 31K in required task space. Hence, the restriction on raster file format is a necessary one to gain this benefit.

4.4 Subframe Image File Formats

Sequential binary files of FIXED record type were selected as the format for the subframe-organized images (and for the stream of MAPS-compressed data). This choice allows use of the DIRECT access mode which in effect gives random image access at the individual record level. Several benefits accrue from this. In the process of conversion from raster to subframe organization, groups of raster lines can be partitioned into partial subframe segments and the corresponding subframe records retrieved and updated individually as needed. An analogous process in the conversion from subframes back to raster lines is also supported by this random access to individual subframe records.

In the adaptive smoothing process, it is necessary to have access to the data from subframes bordering the subframe of current target interest. Again, a random access capability which allows retrieval of individual subframes is required; DIRECT access mode also supports this need.

Finally, some data in the file header is not available until all following records have been completed. The DIRECT access mode allows this first record to be inserted in the file at the end of the process and does so in a simple fashion.

As noted earlier, both DEC file systems - FCS-11 and RMS-11 - support such FIXED record type DIRECT access binary files. Thus, TransMAPS transportability is not compromised.
4.5 COMMENT Guidelines

For a competent FORTRAN programmer, the most useful internal program documentation is the flow of the executable code itself. Once program intent and basic definitions are understood, COMMENTing within the flow can often be more distracting than helpful. Thus, a consistent philosophy of program COMMENTs for TransMAPS has been employed which seeks to aid without intrusion.

File communication, overall intent and structure, and definitions of key variables are grouped at the beginning of each of the seven main modules. Each subroutine, then, is provided with a much briefer heading which also narrates intent and describes local CALLing links. In-line COMMENTs are restricted to setting off major blocks for quick location. On-line flags (using the DEC separator convention '!!') are used to locate points where default values are set and to note points at which I*4 arithmetic is carried out. These flags are supplied to simplify site-specific maintenance and installation changes.

The modularization within TransMAPS is dictated more by the need for logical refinement into 'graspable' chunks than by requirements for code segments which are used repeatedly at different points in the flow. Thus, intermodule communication need not have extensive lists of formal parameters which take on different values at different call points. Rather, the various routines tend to work on a common body of data and use a common set of control parameters. Thus, the communication problem is handled by extensive use of named COMMON blocks. Each main module contains a complete description of the COMMON blocks with variable names, data types, and role characterization. In effect, this description becomes a data dictionary for the module.
The COMMENT formats are summarized schematically in the following tabulation:

**COMMENT Formats:**

---

**Principal Module Headers:**

---

**Program Structure:**

---

**Common Block Communication:**

---

**Subroutine Headers:**

---

**On-line Flags:**

---

**Control Data Corporation - 1982**

<table>
<thead>
<tr>
<th>Files</th>
<th>Unit</th>
<th>Name</th>
<th>Content</th>
<th>From/To Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>In/Out</td>
<td>n</td>
<td>zzzzz descriptor</td>
<td>module SEGMENTED or DIRECT or FIXED SEQ. or FORMATTED</td>
<td></td>
</tr>
</tbody>
</table>

**User Interaction:**

---

**Program Structure:**

---

**Common Block Communication:**

---

<table>
<thead>
<tr>
<th>/blockname/</th>
<th>descriptor length (102 words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>host routine names variable (datatype) descriptor</td>
<td></td>
</tr>
</tbody>
</table>

**Order conventions or geometry definitions (if appropriate):**

---

**Subroutine Headers:**

---

**On-line Flags:**

---

<table>
<thead>
<tr>
<th>Expression</th>
<th>Comment</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>1 Default</td>
<td>(default values set in USERx)</td>
</tr>
<tr>
<td>...</td>
<td>1 I^4</td>
<td>(four-byte integer arithmetic)</td>
</tr>
</tbody>
</table>
SECTION FIVE

TESTING

Verification of the TransMAPS package was divided into three major parts: the MAPS compression logic, the MAPS product formation options, and the annotation capability. This report of the TransMAPS development efforts concludes with a brief discussion of these tests.

5.1 Compression-Logic Diagnostic Test Image

The principal complexities in MAPS arise during the compression process and are further compounded by the extended space of user options in this step. This is the most likely area for small errors in logic or code entry. Moreover, because of the adaptive nature of MAPS, the effects of such errors might remain very localized and easily go undetected in review of the compression of a general image scene. Thus, tests of the compression logic must be capable of exhausting the various patterns of intensity which MAPS might encounter in real imagery. For this purpose, it is sufficient to include only 'generic' patterns which represent all geometries but not all possible intensity levels. A 'binary' image of light (gray scale 0) and dark (gray scale 255) will suffice.

Such a diagnostic test image was created for TransMAPS and is displayed as Figure 5-1. This image is 160 lines by 128 pixels and is suitable for the line-printer pseudo-image display mode of the TransMAPS ANNOTE module. An overlay pattern of lines along 'natural' MAPS boundaries has been added to the pseudo-image.

From the figure it is seen that four complete patterns representing 1x1, 2x2, 4x4, and 8x8 elements are displayed. Also, one quadrant of the pattern at 16x16 and a single dark element at 32x32 have been included. The full patterns at the four lowest resolutions are each made up of twelve 'quads' of light and dark elements. Indeed, all possible
Figure 5-1. MAPS Compression-Logic Diagnostic Image

I-31
arrangements of 'one light and three dark', 'two adjacent light and two adjacent dark', and 'three light and one dark' are present for all four resolution levels. For the 16x16 case, at least one pattern from each of the three classes is represented. Note further that for each resolution, the quad patterns of 'all light' and 'all dark' are shown somewhere in the image.

This test image supports the following classes of verification diagnostics:

**Compression-Logic Diagnostic Tests:**

- Image Partition and Macro-Fidelity Controls
  - Input Image Line and Pixel Skips
  - Subframe Phasing (Square and Staggered)
  - Macro-Partition Group Assignment

**Micro-Fidelity Control:**

- Zigzag Sequencing
- Contrast Control as a Function of Transition Level
- Contrast Control as a Function of Contrast Type
- Pattern Code Assignment

**Gray-Scale Manipulation:**

- Contrast Space Quad Sort
- Intensity Space Quad Sort
- Intensity Reset Assignment

Micro-performance of TransMAPS was verified in all of these areas. The diagnostic test image is also included as part of the TransMAPS package and can be used for verification on installation and, perhaps in an even more valuable role, to familiarize the user with the detailed effects of many of the user options. Several specific examples of its use are given in Section Eight of the User's Manual.
5.2 Product-Generation Familiarization Test Image

Another 'toy' image was generated by resampling the video-sized frame of the IEEE Girl image down to 120 lines by 128 pixels. The result is shown in the pseudo-image of Figure 5-2. The source version was also scaled to six-bit intensity (right-justified in eight bits) to test the six-bit option of the raster to subframe conversion. At this size, the image content is rather strongly undersampled but this makes it effective in emphasizing small artifacts in the resultant product images. The coarse gray-scale granularity of the pseudo-image also serves to enhance otherwise small artifacts. This combination, then, is very suitable for familiarizing the user with the types of fidelity compromises which the MAPS process introduces during compaction.

A series of products were generated to compare various compression and reconstruction strategies with the compression level held constant (at two bits per pixel). This entire sequence is displayed in Section Nine of the User's Manual. In addition to illustrating the strategies, the results of the sequence verify the expected performance characteristics of the various product generation options. This test image is also included in the TransMAPS package. Its small size makes it very effective in exploring a wide range of strategies at very modest computational investment.

5.3 Annotation Option Test Patterns

The various annotation options in ANNOTE were sampled with a small test pattern using the printer pseudo-image for display (see Section Four of the User's Manual). As the final test step, however, a much more complete test of the annotation capability was provided by constructing the test pattern shown in Figure 5-3. This result was generated on a PDP-11/70, transferred to magnetic tape, and then to film using an Optronics Photowrite. The test pattern shows all sixty symbols in all
four orientations at all four magnifications in alternate direct and complement presentations.

This concludes the brief discussion of the development of TransMAPS which constitutes this first part of the project "Final Technical Report." Indeed, this description is transitory and the 'scaffolding' it represents can be removed. The TransMAPS User's Manual is the central portion of this total document and is designed to stand alone in its support of the functional application of the TransMAPS package. Finally, the TransMAPS Maintenance Manual completes the current document. It presupposes understanding and familiarity with the User's Manual and adds the 'systems' perspective necessary to provide transparent software installation and maintenance to the user community.
PREFACE

This is Volume II of the Final Technical Report on Transportable MAPS Software. It constitutes the TransMAPS Software User's Manual; its companion volumes contain a description of TransMAPS development (Volume I) and the TransMAPS Maintenance Manual (Volume III). This volume is submitted in fulfillment of CDRL item A003 of Contract # F30602-80-C-0326.
# TABLE OF CONTENTS

VOLUME II. TRANSMAPS USER'S MANUAL

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>TRANSFORMABLE MAPS SOFTWARE: THE USER'S VIEW</td>
<td>II-1</td>
</tr>
<tr>
<td>1.1</td>
<td>Purpose and Applications</td>
<td>II-1</td>
</tr>
<tr>
<td>1.2</td>
<td>User's Manual Organization</td>
<td>II-2</td>
</tr>
<tr>
<td>1.3</td>
<td>References</td>
<td>II-2</td>
</tr>
<tr>
<td>2.0</td>
<td>TRANSMAPS OVERVIEW</td>
<td>II-4</td>
</tr>
<tr>
<td>2.1</td>
<td>The TransMAPS Package</td>
<td>II-4</td>
</tr>
<tr>
<td>2.2</td>
<td>TransMAPS Structure</td>
<td>II-7</td>
</tr>
<tr>
<td>3.0</td>
<td>TRANSMAPS INTERACTION ENVIRONMENT</td>
<td>II-9</td>
</tr>
<tr>
<td>3.1</td>
<td>Input and Output File Formats</td>
<td>II-9</td>
</tr>
<tr>
<td>3.2</td>
<td>Terminal Interaction Formats</td>
<td>II-10</td>
</tr>
<tr>
<td>3.3</td>
<td>PIP Filename Manipulations</td>
<td>II-12</td>
</tr>
<tr>
<td>4.0</td>
<td>IMAGE ASSEMBLY AND ANNOTATION</td>
<td>II-15</td>
</tr>
<tr>
<td>4.1</td>
<td>Annotation and Image Assembly Planning Form</td>
<td>II-15</td>
</tr>
<tr>
<td>4.2</td>
<td>Output Frame Specification</td>
<td>II-17</td>
</tr>
<tr>
<td>4.3</td>
<td>Embedded Input Image Specification</td>
<td>II-18</td>
</tr>
<tr>
<td>4.4</td>
<td>Annotation Message Specifications</td>
<td>II-19</td>
</tr>
<tr>
<td>4.5</td>
<td>Image Labeling Example</td>
<td>II-21</td>
</tr>
<tr>
<td>4.6</td>
<td>Annotation Options Examples</td>
<td>II-23</td>
</tr>
<tr>
<td>5.0</td>
<td>MAPS COMPRESSION/DECOMPRESSION: BASIC USE</td>
<td>II-28</td>
</tr>
<tr>
<td>5.1</td>
<td>Raster to Subframe Conversion - SUBFRM.</td>
<td>II-28</td>
</tr>
<tr>
<td>5.2</td>
<td>MAPS Compression - MAPS</td>
<td>II-29</td>
</tr>
<tr>
<td>5.3</td>
<td>MAPS Decompression - DMAPS</td>
<td>II-29</td>
</tr>
<tr>
<td>5.4</td>
<td>Subframe to Raster Conversion - RASTER</td>
<td>II-30</td>
</tr>
<tr>
<td>5.5</td>
<td>User Interaction Protocol ('Quick' Mode)</td>
<td>II-30</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td>MAPS CONCEPTS AND PROCESSES</td>
<td>II-37</td>
</tr>
<tr>
<td>6.1</td>
<td>Image Partitioning</td>
<td>II-37</td>
</tr>
<tr>
<td>6.2</td>
<td>Sequence Conventions</td>
<td>II-38</td>
</tr>
<tr>
<td>6.3</td>
<td>Micro-fidelity Control</td>
<td>II-41</td>
</tr>
<tr>
<td>6.4</td>
<td>MAPSel Coding</td>
<td>II-44</td>
</tr>
<tr>
<td>6.5</td>
<td>Adaptive 'Convolution'</td>
<td>II-47</td>
</tr>
<tr>
<td>7.0</td>
<td>MAPS USER OPTIONS</td>
<td>II-50</td>
</tr>
<tr>
<td>7.1</td>
<td>Raster to Subframe Conversion</td>
<td>II-51</td>
</tr>
<tr>
<td>7.2</td>
<td>MAPS Compression</td>
<td>II-51</td>
</tr>
<tr>
<td>7.2.1</td>
<td>User Interaction Mode</td>
<td>II-52</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Macro-Fidelity Control</td>
<td>II-54</td>
</tr>
<tr>
<td>7.2.3</td>
<td>Micro-Fidelity Control</td>
<td>II-54</td>
</tr>
<tr>
<td>7.2.4</td>
<td>Gray-Scale Manipulations</td>
<td>II-55</td>
</tr>
<tr>
<td>7.3</td>
<td>MAPS Decompression and Resolution Image</td>
<td>II-57</td>
</tr>
<tr>
<td>7.4</td>
<td>MAPS Adaptive Smoothing</td>
<td>II-58</td>
</tr>
<tr>
<td>7.5</td>
<td>MAPS Difference Image Formation</td>
<td>II-58</td>
</tr>
<tr>
<td>7.6</td>
<td>Subframe to Raster Conversion</td>
<td>II-59</td>
</tr>
<tr>
<td>8.0</td>
<td>MAPS COMPRESSION: EXTENDED USE</td>
<td>II-60</td>
</tr>
<tr>
<td>8.1</td>
<td>MAPS Compression Planning Form</td>
<td>II-60</td>
</tr>
<tr>
<td>8.2</td>
<td>TransMAPS Compression Diagnostic Test Image</td>
<td>II-60</td>
</tr>
<tr>
<td>8.3</td>
<td>User Interaction Protocol ('User' and 'Full' Modes)</td>
<td>II-64</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>9.0</td>
<td>MAPS PRODUCT GENERATION: EXTENDED USE</td>
<td>II-78</td>
</tr>
<tr>
<td>9.1</td>
<td>MAPS Product Generation Planning Form</td>
<td>II-78</td>
</tr>
<tr>
<td>9.2</td>
<td>User Interaction Protocol (All Modules)</td>
<td>II-78</td>
</tr>
<tr>
<td>9.3</td>
<td>Resolution (level) Image</td>
<td>II-88</td>
</tr>
<tr>
<td>9.4</td>
<td>Block Decompression</td>
<td>II-88</td>
</tr>
<tr>
<td>9.5</td>
<td>Block Mode with Uniform Adaptive Smoothing</td>
<td>II-88</td>
</tr>
<tr>
<td>9.6</td>
<td>Pattern Decompression</td>
<td>II-89</td>
</tr>
<tr>
<td>9.7</td>
<td>Pattern Mode with Uniform Adaptive Smoothing</td>
<td>II-89</td>
</tr>
<tr>
<td>9.8</td>
<td>Pattern Mode with Gaussian Adaptive Smoothing</td>
<td>II-89</td>
</tr>
<tr>
<td>9.9</td>
<td>Pattern/Gaussian Mode with Dither</td>
<td>II-89</td>
</tr>
<tr>
<td>9.10</td>
<td>Mean Square Error Performance Comparisons</td>
<td>II-105</td>
</tr>
<tr>
<td>9.11</td>
<td>A 'Real'-Image Example</td>
<td>II-105</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>FULL SIZE TransMAPS PLANNING FORMS</td>
<td>II-113</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>TransMAPS Structure</td>
<td>II- 8</td>
</tr>
<tr>
<td>4-1</td>
<td>Annotation and Image Assembly Planning Form</td>
<td>II-16</td>
</tr>
<tr>
<td>4-2</td>
<td>Image Labeling Planning Form</td>
<td>II-22</td>
</tr>
<tr>
<td>4-3</td>
<td>Example of ANNOTE Image Labeling</td>
<td>II-24</td>
</tr>
<tr>
<td>4-4</td>
<td>Annotation Example Planning Form</td>
<td>II-25</td>
</tr>
<tr>
<td>4-5</td>
<td>Pseudo-image Annotation Examples</td>
<td>II-26</td>
</tr>
<tr>
<td>4-6</td>
<td>Annotation Options</td>
<td>II-27</td>
</tr>
<tr>
<td>5-1</td>
<td>Basic MAPS Compression Example</td>
<td>II-36</td>
</tr>
<tr>
<td>6-1</td>
<td>MAPS Image Partition Concepts</td>
<td>II-39</td>
</tr>
<tr>
<td>6-2</td>
<td>MAPS Sequence Concepts</td>
<td>II-40</td>
</tr>
<tr>
<td>6-3</td>
<td>MAPS Contrast Control</td>
<td>II-42</td>
</tr>
<tr>
<td>6-4</td>
<td>MAPS Threshold Selection</td>
<td>II-43</td>
</tr>
<tr>
<td>6-5</td>
<td>MAPS Pattern Mode</td>
<td>II-46</td>
</tr>
<tr>
<td>6-6</td>
<td>MAPS Adaptive 'Convolution'</td>
<td>II-48</td>
</tr>
<tr>
<td>7-1</td>
<td>MAPS Compression Options Overview</td>
<td>II-53</td>
</tr>
<tr>
<td>8-1</td>
<td>MAPS Compression Planning Form</td>
<td>II-61</td>
</tr>
<tr>
<td>8-2</td>
<td>MAPS Compression-Logic Diagnostic Image</td>
<td>II-62</td>
</tr>
<tr>
<td>8-3</td>
<td>Planning Form for MTEST Example</td>
<td>II-65</td>
</tr>
<tr>
<td>8-4</td>
<td>MAPS Decompressed and Resolution Images for Diagnostic Test Example</td>
<td>II-77</td>
</tr>
<tr>
<td>9-1</td>
<td>MAPS Product Generation Planning Form</td>
<td>II-79</td>
</tr>
<tr>
<td>9-2</td>
<td>MAPS Resolution Image at 2 bits/pixel</td>
<td>II-90</td>
</tr>
<tr>
<td>9-3</td>
<td>Simple Averaging to Achieve 2 bits/pixel</td>
<td>II-91</td>
</tr>
<tr>
<td>9-4</td>
<td>Error Histogram for 2x2 MEAN Example</td>
<td>II-92</td>
</tr>
<tr>
<td>9-5</td>
<td>MAPS Block Mode at 2 bits/pixel</td>
<td>II-93</td>
</tr>
<tr>
<td>9-6</td>
<td>Error Histogram for DMAPS B2 Block Mode Example</td>
<td>II-94</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>9-7</td>
<td>MAPS Block Mode with Adaptive Smoothing (Uniform Weight)</td>
<td>II-95</td>
</tr>
<tr>
<td>9-8</td>
<td>Error Histogram for ADAPT B2 Block Mode Example</td>
<td>II-96</td>
</tr>
<tr>
<td>9-9</td>
<td>MAPS Pattern Mode at 2 bits/pixel</td>
<td>II-97</td>
</tr>
<tr>
<td>9-10</td>
<td>Error Histogram for DMAPS P2 Pattern Mode Example</td>
<td>II-98</td>
</tr>
<tr>
<td>9-11</td>
<td>MAPS Pattern Mode with Adaptive Smoothing (Uniform Weight)</td>
<td>II-99</td>
</tr>
<tr>
<td>9-12</td>
<td>Error Histogram for ADAPT P2 Pattern Mode Example</td>
<td>II-100</td>
</tr>
<tr>
<td>9-13</td>
<td>MAPS Pattern Mode with Adaptive Smoothing (Gaussian Weight)</td>
<td>II-101</td>
</tr>
<tr>
<td>9-14</td>
<td>Error Histogram for GAUSS P2 Pattern Mode Example</td>
<td>II-102</td>
</tr>
<tr>
<td>9-15</td>
<td>MAPS Pattern/Gaussian Mode with Dither</td>
<td>II-103</td>
</tr>
<tr>
<td>9-16</td>
<td>Error Histogram for Pattern/Gaussian DITHER 8 Example</td>
<td>II-104</td>
</tr>
<tr>
<td>9-17</td>
<td>Compression Planning Form - Building Scene</td>
<td>II-106</td>
</tr>
<tr>
<td>9-18</td>
<td>Product Generation Planning Form - Building Scene</td>
<td>II-107</td>
</tr>
<tr>
<td>9-19</td>
<td>Annotation Planning Form - Building Composite</td>
<td>II-108</td>
</tr>
<tr>
<td>9-20</td>
<td>'Real'-Image MAPS Example - Building Scene</td>
<td>II-110</td>
</tr>
<tr>
<td>9-21</td>
<td>Error Histogram for Building Scene</td>
<td>II-111</td>
</tr>
</tbody>
</table>
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Contents of TransMAPS</td>
<td>II-5</td>
</tr>
</tbody>
</table>
SECTION ONE

TRANSPORTABLE MAPS SOFTWARE: THE USER'S VIEW

This document provides formal description of the Transportable MAPS Software Package or 'TransMAPS' from a user's viewpoint.

1.1 Purpose and Applications

Micro-Adaptive Picture Sequencing (MAPS) is a computationally-efficient, contrast-adaptive, variable-resolution spatial image coding technique. The TransMAPS Software Package implements the MAPS processes and related support functions in an integrated software system which is designed to be transportable among a variety of high-use mini-computers in the DEC computer family. The purpose of this implementation is to broaden access to MAPS. The ultimate intent is to establish a vehicle suitable for direct exploration of the MAPS technique and to provide a system capable of supporting functional application of MAPS to real image coding tasks.

The current document addresses those areas specifically concerned with the user's application of the TransMAPS software. These include an overview of the structure of TransMAPS, general guidelines for user interaction, detailed information on user options, a concise description of the underlying MAPS concepts and processes, and several examples of MAPS interactive protocols. The intended audience includes both those who wish to use TransMAPS as a 'black box' for operational image coding and those who wish to explore the MAPS technique itself. The emphasis here is on 'what it does' and 'how to invoke it'.

II-1
1.2 User's Manual Organization

Sections Two through Five provide the information needed to gain an initial facility with the basic TransMAPS capability. Section Two presents an overview of the contents and structure of the package. Section Three discusses general guidelines for the TransMAPS interaction environment. Section Four then describes the stand-alone image assembly and annotation module, ANNOTE. This module provides rudimentary but rapid and effective image display support, even for systems with no formal image display device. Finally, Section Five describes the basic protocols needed for MAPS compression and decompression operations.

Sections Six through Nine cover topics intended to give the user much deeper understanding and fluency in applying TransMAPS. Section Six presents a summary of the underlying MAPS concepts and processes. Section Seven then describes the entire space of interactive user options in detail and on a module-by-module basis. Finally, Sections Eight and Nine present protocols for the extended use of the TransMAPS compression and product generation capabilities.

1.3 References

The information in this manual provides a self-contained guide to the use of TransMAPS. However, the related volumes - TransMAPS Final Technical Report and TransMAPS Maintenance Manual - may be helpful in giving additional context on the development of the package and its specific embodiment of the MAPS processes.

For more detailed information on the origin and evolution of MAPS, the user may also wish to consult the following reports and articles:
References:

RADC MAPS-Related Reports:


SPIE Proceedings Articles:


SECTION TWO

TRANSMAPS OVERVIEW

This section gives an overview of the contents and structure of TransMAPS.

2.1 The TransMAPS Package

The contents of the TransMAPS Package are summarized in Table 2-I. The installed software consists of seven process modules supported by an extended MAPS file structure and several related test image and data files.

The first six program modules are all directly MAPS-related. This group is further subdivided into modules concerned with MAPS compression (#1 and #2) and modules concerned with MAPS product formation (#3, #4, #5, and #6). The seventh module provides a stand-alone image assembly and annotation capability which also gives immediate image display support for the MAPS processes.

Modules #1 (SUBFRM) and #6 (RASTER) embody the basic image interface processes - conversion between external raster image format and internal MAPS subframe organization. Modules #2 (MAPS), #3 (DMAPS), #4 (ADAPT), and #5 (DIFFER) operate on subframe-organized imagery. They provide respectively, MAPS compression, MAPS decompression and resolution image formation, MAPS adaptive image smoothing, and MAPS difference image formation. Module #5 also yields an evaluation of MAPS fidelity performance. Finally, module #7 assembles up to two input raster images into a single frame and allows addition of annotation in a variety of orientations and type sizes. The resultant output image can be either in the form of another standard binary raster file or in the form of a formatted pseudo-image file suitable for listing on the system line printer.
TABLE 2-I. CONTENTS OF TRANSMAPS.

**TransMAPS Package:**

<table>
<thead>
<tr>
<th>Seven Main Program Modules:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>#1</strong> SUFRMM Raster to Subframe Conversion</td>
</tr>
<tr>
<td><strong>#2</strong> MAPS MAPS Compression</td>
</tr>
<tr>
<td><strong>#3</strong> DMAPS MAPS Decompression &amp; Level Image Formation</td>
</tr>
<tr>
<td><strong>#4</strong> ADAPT MAPS Adaptive Image Smoothing</td>
</tr>
<tr>
<td><strong>#5</strong> DIFFER MAPS Difference Image Formation</td>
</tr>
<tr>
<td><strong>#6</strong> RASTER Subframe to Raster Conversion</td>
</tr>
<tr>
<td><strong>#7</strong> ANNOTE Image Assembly and Annotation</td>
</tr>
</tbody>
</table>

**Provided on Computer-Compatible Tape (CCT):**

- FORTRAN IV-Plus Source Code - x.FSV
- FORTRAN Object (F4P) Code - x.OBJ

**Six Data Sets:**

<table>
<thead>
<tr>
<th>Annotation Symbol-Map Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPS Compression Test Image (160 x 128)</td>
</tr>
<tr>
<td>Sample MAPS User Parameter Set (Use with MTEST)</td>
</tr>
<tr>
<td>MAPS Product Generation Test Image (120 x 128)</td>
</tr>
<tr>
<td>Two ‘video Frame’ Images (480 lines x 524 pixels):</td>
</tr>
<tr>
<td>- Building Scene</td>
</tr>
<tr>
<td>- IEEE Girl</td>
</tr>
</tbody>
</table>

**Provided on Computer-Compatible Tape (CCT):**

**MAPS File Structure with Standard Filenames and Headers:**

| Source Image (One Subframe/FIXED Record) | - IMAGE.DAT |
| User Parameter Set | - MSET.DAT |
| MAPS Compression Stream (FIXED Records) | - MAPS.DAT |
| MAPS Block/Pattern Image (Subframes) | - DMAPS.DAT |
| MAPS Resolution Image (Subframes) | - LEVEL.DAT |
| MAPS Adaptively Smoothed Image (Subframes) | - ADAPT.DAT |
| MAPS Difference Image (Subframes) | - ERROR.DAT |
| Fidelity Performance Summary (Listing) | - EPRINT.DAT |
| MAPS Product Image (Raster) | - X.RAS.DAT |
| x = I,D,L,A,E |
| Annotation Symbol-Map Tables | - SYMBOL.BIN |
| Annotated Image (Raster) | - ANIMG.DAT |
| Annotated Printer Pseudo-Image (Listing) | - APRINT.DAT |

**User Interaction Pre-Planning Aids:**

- MAPS Planning Form - Compression
- MAPS Planning Form - Product Generation
- Annotation and Image Assembly Planning Form

**Documentation:**

- TransMAPS User’s Manual
- TransMAPS Maintenance Manual
Six data sets are provided with the package. The first contains the bit-map tables for the annotation symbol set. This resides as a system data file, SYMBOL.DAT, and is read in automatically by module #7, ANNOTE when it is invoked. The second and third data sets consist of a 'toy'-sized diagnostic test image, MTEST.DAT, and a corresponding sample set of user parameters, MSET.DAT. The fourth data set is also a 'toy'-sized test image, GIRL6.DAT; it is particularly suitable for initial familiarization with the MAPS processes. The fifth and sixth data sets, BLDGIMG.DAT and GIRLIMG.DAT, are examples of real-world 'video'-sized images.

Communication of image data among the modules is provided through the MAPS file structure. Ancillary data is contained in standardized file headers and each file type has a unique but standard name. Thus, the intermodule data transfer is transparent to the user. However, the file names provide simple mnemonics for their contents so the user can assess the status of MAPS processing through a simple review of the appropriate file directory.

Application of TransMAPS is supported externally by a group of user aids and the set of software documentation. The aids are presented as three 'planning forms' for MAPS Compression, MAPS Product Generation, and Annotation and Image Assembly. These forms chart the extensive space of MAPS user options and allow its structure to be seen at a glance. User entries on the forms provide for both pre-planning and documentation of TransMAPS interactive sessions.

The formal software documentation consists of the TransMAPS Maintenance Manual and this TransMAPS User's Manual.
2.2 TransMAPS Structure

The structure of the TransMAPS software system is depicted in Figure 2-1. This presentation shows the detailed relationship between the seven TransMAPS process modules and the TransMAPS system of MAPS standard files. Figure 2-1 provides a self-contained roadmap to TransMAPS and should be viewed as the key reference whenever a system overview is required.
Figure 2-1. TransMAPS Structure
SECTION THREE

TRANSMAPS INTERACTION ENVIRONMENT

TransMAPS is designed to be extensively interactive to accommodate the large space of available user options. This section describes the general guidelines appropriate to these interactions.

3.1 Input and Output File Formats

The communication of external image data files to and from the TransMAPS system takes place in modules #1 (SUBFRM), #6 (RASTER) and #7 (ANNOTE). TransMAPS expects these images in the form of raster-organized binary files with one image line per logical record. The raster geometry conventions require the pixel index to increase sequentially along each line from left to right, and the line index to increase sequentially down the image from top to bottom. This is essentially a video convention for the geometry except that it does not involve line interlace.

Two pixel formats are supported for the external imagery in TransMAPS - an eight-bit pixel and six-bit pixel right-justified in an eight-bit field (the so-called DICOMED format).

TransMAPS is designed to accommodate logical records up to 4000 pixels (bytes) in length, although such large images are computationally expensive to process on a mini-computer system. However, in order to minimize buffer space requirements, TransMAPS expects the logical records to be partitioned into much smaller physical records through the use of the SEGMENTED record type. Since no universal standard for image file format is available, each local 'standard' digital image form is unknown a priori and likely to vary from site to site. Thus, responsibility for expressing the raster source imagery as a SEGMENTED binary file is...
presumed to reside with personnel at each site. The test imagery supplied with the TransMAPS package is, of course, already in this SEGMENTED format.

3.2 Terminal Interaction Formats

Each of the seven TransMAPS modules is invoked by an MCR (DEC's Monitor Console Routine) command of the generic form 'RUN Taskname'. Here, 'Taskname' takes on either a long or short version for each module as listed in the following tabulation:

| Module #1: | SUBFRM or SF |
| Module #2: | MAPS or MP |
| Module #3: | DMAPS or DM |
| Module #4: | ADAPT or AD |
| Module #5: | DIFFER or DF |
| Module #6: | RASTER or RS |
| Module #7: | ANNOTE or AI |

Once the module is invoked, code internal to the module directs the interaction to select values or states for the required user options. Typically, the program requests an update to an option with a query in the form:

'Option descriptor' ? ('Allowed Range') 'Current Value'

The 'Allowed Range' is adjusted dynamically to account for any changes in constraint imposed by prior option selections. 'Current Value' initially displays the system default value for the option. Thereafter, it presents either the most recent selection or the closest value from the current allowed range.
Insofar as possible, user responses to these option queries are limited to a very small number of generic types. Consistent response formats have been sought. These types divide first into two general categories - numeric and alpha-literal. The numerics are mostly single numbers (a string of contiguous decimal digits, possibly with sign and decimal point). Occasionally, a numeric response in the form of a vector of numbers is required. In the vector case, successive components are separated by simple blank spaces. If several adjacent components have the same value, they may be entered with the sequence of a 'repeat count' followed by an asterisk followed by the common 'value' (r*v). For all numerics, a slash character (/) followed by a carriage return is used to denote the default response, 'no change'. A slash (/) inserted before a list of vector components is exhausted denotes that all remaining components are left unchanged.

The alpha-literal responses further subdivide into three types - Y or N for 'yes' or 'no'; a single mnemonic letter for a menu choice; and a contiguous symbol string for a filename, an image name, or a message text. The 'no change' default for alpha-literals is normally a simple carriage return. However, for consistency with the numeric case, a leading slash (/) is also interpreted as the 'no change' response. The only limitation that this imposes is that a message text cannot begin with a slash symbol (/).

Module #2 (MAPS) and module #7 (ANNOTE) have particularly extensive option spaces. In order to avoid the tedium of providing long strings of 'no change' responses, the interactions for these modules have been subdivided into smaller groups of related options. At the beginning of each group, the user is given the choice of entering the group for option review and update or simply accepting the current values for all options in the group.
The user is also given extensive recovery and change control in the interaction. For matrix option specifications such as those for macro-fidelity image partition or micro-fidelity contrast thresholds, the code provides immediate feedback by updating and displaying the matrix after each line (vector) of user responses. Moreover, the user can access any line again and in any order to make corrections. In effect, a simple 'editing' mode is available. Similarly, for extended text entry such as that which may be encountered in defining annotation messages, all text is collected and displayed together on the terminal screen after initial entry. Again, an 'editing' mode is enabled which allows individual messages from this collection to be repeatedly corrected until the total text is satisfactorily defined.

Finally, the last query in the interaction for each module gives the user the choice of returning to review (and possibly modify) the entire set of option selections or to proceed. This insures the opportunity for parameter verification before the image processing resources are committed.

3.3 PIP Filename Manipulations

The modular structure of TransMAPS allows many strategies and time sequences for the application of the package. For example, the raster to subframe conversion (SUBFRM) might be run once on a particular source image. Next, the MAPS compression process (MAPS) might be run several times with different control parameters before proceeding with MAPS product generation. Each invocation of a process module produces a new set of output files. For files with the same filename - for example, MAPS.DAT - successive files are distinguished by increasing file 'version numbers'. The file specification has the form MAPS.DAT;'version' where 'version' is an octal number. When a module is invoked which requires a
particular file type as input - for example, module #3 DMAPS requires MAPS.DAT - it automatically opens and accesses the latest version.

In order to access earlier versions of a file, some capabilities of DEC's PIP utility (Peripheral Interchange Program) can be exploited. In particular, the PIP switches '/RE', '/EN', and '/RM' along with the subswitch '/NV' are relevant. These switches play the following roles:

/RE  Rename - allows any portion of a file specification, 'filename.filetype;version' to be changed;

/EN  Enter - allows a synonym file specification to be entered into the file directory, access to the file is allowed under any of its synonyms;

/RM  Remove - allows a synonym to be removed from the file directory without removing the file or its other names;

/NV  New Version - supplies a 'version number' one larger than the latest version when used with /RE or /EN.

An example of the use of these switches is provided by the following generic command line:

    PIP Filename.DAT/RE/NV = Filename.DAT;earlyversion.

In this case, 'earlyversion' is renamed to the (new) latest version and will be accessed by any module which opens 'Filename.DAT' for input.
Additional PIP facilities which are useful in controlling file proliferation are the purge and delete switches '/PU' and '/DE'. PURGE eliminates all but the most recent version associated with a particular filename and DELETE eliminates a file with an explicit version number specified. Obviously, these commands must be used with care to avoid inadvertent deletion of files which were to be kept.

These various PIP system utilities clearly provide great flexibility in maintaining the MAPS file environment.
SECTION FOUR

IMAGE ASSEMBLY AND ANNOTATION

Module #7, ANNOTE, provides a stand-alone image assembly and annotation capability which can be used either with MAPS products output from module #6 or with other user-supplied raster images. Because of its role in 'quick-look' pseudo-image display (using the system line printer), ANNOTE is introduced in this section, out of normal order. As a consequence, ANNOTE becomes available to provide visual output for examination of the results of MAPS process applications.

4.1 Annotation and Image Assembly Planning Form

The user options for ANNOTE are summarized in the 'Annotation and Image Assembly Planning Form' presented in reduced size as Figure 4-1. The actual form just fills the length of an 8-1/2" by 11" sheet - large enough for comfortable user entry. ANNOTE allows the assembly of up to two input raster images into a single output frame. Moreover, it can be run recursively to assemble several images by taking the output from a previous run as one of the inputs to the current run. Restriction to two input images at a time is largely a consequence of the 32K task size limit imposed by the sixteen-bit word length of the PDP-11.

ANNOTE has two output frame modes. In the first, an ordinary binary raster image file is created. This is the mode to be used for recursion or for making large image frames (up to 4000 pixels) for subsequent transfer to a large-format display device such as an Optronics Photowrite. The second mode is limited to image widths of 128 pixels per line and creates a formatted pseudo-image file to be listed on the system line printer. This mode has very coarse intensity granularity (only eight gray-scale levels) and uses only one overprint per line. Even with
### Annotation and Image Assembly Planning Form

<table>
<thead>
<tr>
<th>Type (Raster file)</th>
<th>Printer listing</th>
<th>Lines</th>
<th>Pixels</th>
<th>Complement (Y N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMENS.DAT</td>
<td>APRINT.DAT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Embedded Images:

<table>
<thead>
<tr>
<th>Image</th>
<th>Filename</th>
<th>Input Position</th>
<th>Output Position</th>
<th>Size</th>
<th>Pixels</th>
<th>Complement (Y N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Embedded Messages:

<table>
<thead>
<tr>
<th>Line</th>
<th>Symbol</th>
<th>Center</th>
<th>Text 1</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
<th>40</th>
<th>45</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>(TMkR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4-1. Annotation and Image Assembly Planning Form
these severe dynamic range restrictions, the resultant 'images' convey a surprising amount of information. Indeed, certain small irregularities and artifacts are enhanced by the process so it makes a very effective tool for exploring MAPS fidelity performance.

Two of the main features of the ANNOTE option space have just been discussed. From the planning form in Figure 4-1, however, it is seen that there are three groups of options and each group contains several entries. These three categories will be discussed in the next three subsections.

4.2 Output Frame Specification

The first user option is the choice between output image types - raster file (the default) or printer pseudo-image. The standard filename for the raster file is ANIMG.DAT; the filename for the printer listing is AFRINT.DAT.

The next pair of options specify the size of the output frame. Numeric values for the number of image LINES and the number of PIXELS per line are required here. The number of bits per pixel is always assumed to be eight in this module. Note that the 'video' geometry conventions - pixel index increasing left to right, line index increasing top to bottom - are in force for all images in ANNOTE.

The final output frame option is the choice of whether to make the frame background clear (gray-scale 0) or opaque (gray-scale 255). The default is 'clear' but a 'Y' response to the 'COMPLEMENT ?' query will change the background to 'opaque'.
4.3 Embedded Input Image Specifications

The first option is the choice of the number of input images to be assembled. The allowed values are 0, 1, and 2. A value of '0' skips the rest of the specification and would be chosen if a 'text-only' output frame were desired. A value of '1' would be chosen if it were desired to annotate a single image. A value of '2' implies the assembly of a pair of images, one or both of which might have been the output from previous ANNOTE invocations.

Each embedded input image then requires specification of eight additional selections. First, the filename for the input image must be entered; this can be up to nine characters long and is an alpha-numeric string. If the input image is a MAPS raster product, the filename will be one of IRAST, DRAST, LRAST, ARAST, or ERAST dependent on the product type. The latest version of the corresponding file will be accessed. Raster images other than MAPS products, or MAPS images which have been renamed using the PIP utilities, will have to have the corresponding filenames entered as they appear in the file directory. Note that in all cases, the file type must be '.DAT' and is not entered explicitly.

The next two input image parameters involve file positioning. ANNOTE contains provision for skipping into the image by a specified number of lines and pixels. The queries require numeric values for SKIP LINES and SKIP PIXELS. The default for each is zero.

The size of the embedded image is selected next through input of two numerics for LINES and PIXELS. Note that the embedded image must be smaller than the output frame; the program enforces this condition if
values which exceed this are transmitted. The number of pixels allowed is also constrained by the condition:

\[
\text{PIXELS} + \text{SKIP PIXELS} < 4001
\]

Two more numerics are required to specify the position of the embedded input image in the output frame. The position is given in terms of the output frame location for the START LINE and START PIXEL corresponding to the upper left pixel ('origin') of the input image. Here again, the allowed range is determined and displayed in terms of the relative sizes of the input and output frames; violations are automatically reset to the closest allowed value.

The final option for each input image is the choice of whether the gray scale is to be direct (the default) or COMPLEMENTED. This feature is useful in creating or converting images in 'negative' form.

Note that when two input images are to be assembled, they are allowed to 'conflict' within the output frame even though each must fit individually. The conflict resolution convention is that Input Image 2 overwrites Input Image 1 where they overlap. Thus, if successive images are being added to a frame by recursion, the output image from the previous ANNOTE run should be Input Image 1 in the current run.

4.4 Annotation Message Specifications

The number of annotation messages can range from zero to twenty and this is the first option selected in the embedded annotation interaction. If zero is chosen, the remainder of the interaction is skipped.

Each message, up to the number chosen, is then entered in a hierarchical fashion. Successive option choices for the message tend to constrain
later choices and this is reflected in the dynamic changes of the respective 'allowed range' displays. Seven options must be entered for each message.

Messages can be oriented in any of four directions with the top of the symbols toward the 'Top', 'Bottom', 'Left', or 'Right' of the output frame. This facility is provided so that the annotation can be matched to the scene content if the scene doesn't match the 'video' geometry. The user must respond with a single character literal T, B, L, or R to this orientation query.

Once the orientation has been specified, the number of symbols of each size which can be fit within the output frame can be determined. The next option requires the user to give a numeric specification of the message length in characters. This length can be up to the maximum allowed by the image frame at the desired symbol size (but no more than fifty characters if that is smaller). Allowed ranges as a function of symbol size are supplied as part of the interactive user prompt for this option selection. Note that a particular message can be deleted during any editing step by simply specifying a zero character count.

The symbol size is then chosen from those still allowed for this orientation and message length. Initially, symbol sizes of 1x, 2x, 3x, and 4x are possible. These correspond to characters in frames which are 16x16, 32x32, 48x48, and 64x64 pixels in size.

Location of the message in the output frame is selected next. This position is specified by giving the output frame CENTER LINE and CENTER PIXEL coordinates. Specification of the position of the center of the message avoids having to remember an orientation-dependent convention.

The user is also given the choice of whether the characters are to be
'direct' (opaque symbols in a clear frame) or COMPLEMENTED (clear symbols in an opaque frame).

Finally, the user supplies the message text stream itself. This is prompted with a display of the sixty allowed symbols in the annotation character set and a line of dashes corresponding to the chosen length of the message.

Note that annotation overwrites the embedded images where they overlap. Later messages can also overwrite earlier messages. However, this is usually not desired except in very special circumstances. Thus, ANNOTE provides a message conflict analysis and prompts editing of either the overwritten or overwriting message to correct the situation. Nevertheless, the conflict can be retained if desired.

4.5 Image Labeling Example

Figure 4-2 shows a completed planning form for a simple example of image labeling. The pseudo-image output mode has been chosen and a single input image is used. This image is the GIRL6.DAT 'toy' test image which has been converted from 6-bits to 8-bits by running it successively through SUBFRM and RASTER. Note also that the product image IRAST.DAT has been renamed to GIRL.DAT using PIP with a /RE switch.

The output frame was given twenty more lines than the input frame size. This extra space was then used to 'annotate' the input image with a message outside of its boundaries. The 'message' in this case simply lists the image size in 'lines x pixels'; the smallest symbol size was chosen here.

The results of the ANNOTE interactive session are displayed as Figure 4-3. Note here that the printer has been reset to II-21.
### Image Labeling Planning Form

**RUN ANNOTATE**

**OUTPUT IMAGE:**
- Type: Printed Listing
- Lines: 140
- Pixels: 128
- Complement: Yes

**EMBEDDED IMAGES:**
- Number: 0

**EMBEDDED MESSAGES:**
- Number: 0

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>CENTER</th>
<th>PIXEL</th>
<th>COMP</th>
<th>TEXT 1</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
<th>40</th>
<th>45</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HI</td>
<td>190</td>
<td>64</td>
<td>(y)</td>
<td>120x128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td>(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4-2. Image Labeling Planning Form
eight-lines-per-inch mode to make the 'pixels' more nearly square (a ratio of 8:10 rather the normal printer 6:10). This printer mode is recommended for consistent use with ANNOTE.

4.6 Annotation Options Examples

Figure 4-4 presents the completed planning form for an 'annotation-only' output frame which exhibits samples of the various annotation options. Again, the pseudo-image output form was selected but this time with the background COMPLEMENTED. The messages were constructed to show the range of symbol size, the four message orientations, direct and COMPLEMENTED messages, and the complete set of sixty characters.

The resultant printer listing is displayed as Figure 4-5. The various messages fill the output frame except for a small rectangular patch in the upper left-hand corner; there the effect of the background complement is seen. Independent planning and replication of this image is an excellent exercise to familiarize the new user with the annotation process.

A much more ambitious illustration of the annotation options is portrayed in Figure 4-6. Here, all sixty symbols appear in all four sizes at all four orientations with alternate direct and COMPLEMENT intensities. This example was prepared on a PDP-11/70 and then transferred via magnetic tape to an off-line Optronics Photowrite facility. Figure 4-6 summarizes how the ANNOTE annotation options appear in a real image display mode.
Figure 4-3. Example of ANNOTE Image Labeling
**Figure 4-4. Annotation Example Planning Form**

<table>
<thead>
<tr>
<th>FRAME</th>
<th>FILLER</th>
<th>LINE</th>
<th>PIXELS</th>
<th>OUTPUT POSITION</th>
<th>START LINE</th>
<th>START PIXEL</th>
<th>COPIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 4-5. Pseudo-image Annotation Examples
SECTION FIVE

MAPS COMPRESSION/DECOMPRESSSION: BASIC USE

The basic or core MAPS processes are contained in four modules - #1 SUBFRM, #2 MAPS, #3 DMAPS, and #6 RASTER. This section describes the broad classes of user interaction required by each of these modules. It then illustrates the entire flow by presenting the complete interactive protocol for a simple example. The protocol is a direct photocopy of the resulting DEC-writer listing.

5.1 Raster to Subframe Conversion - SUBFRM

The user interaction in module #1 is divided into four option groups. The first group involves 'source image indentification' and requires specification of the filename for the source raster and a user image name to be carried in the MAPS standard file headers.

The second group involves 'source image position specification' and requires input of the number of lines and pixels to be skipped into the source raster file. The defaults are zero for both lines and pixels.

The third group involves 'source image size specification' and requires input of the number of lines, the number of pixels per line, and the number of bits per pixel for the portion of the source raster frame to be processed. The number of pixels to be skipped plus the number of pixels retained can total up to 4000. The number of bits per pixel can be either eight or six right-justified in eight; the default is eight.

The final group involves 'source image partition specification' and requires choice of the subframe size to be used. The subframes can be 8x8, 16x16, or 32x32 pixels. The source image frame is automatically
padded in both the line and pixel directions to allow division into an integral number of complete subframes. If a subframe size of 8x8 is chosen, the user has the additional option of selecting between a square grid of subframes or a 'staggered' grid. In the staggered case, each successive subframe along the pixel direction starts three lines later (or five lines earlier) than the immediately preceding subframe. This gives the grid somewhat of a 'brick wall' appearance and tends to break up the 'blockiness' of the partition. The default selections are 8x8 subframes in a square grid.

5.2 MAPS Compression - MAPS

The complete option space for module #2 allows subtle and flexible control over the MAPS compression process. However, for basic MAPS compression applications, a very simple control strategy suffices. The first query in the interaction for this module requires selection of the subsequent interactive mode. For routine use, the 'Quick' mode should be selected. The other two modes, 'User' and 'Full', will be elaborated in Section Seven. Note that the 'Quick' mode is also the default.

The only other option required of the user, then, is specification of the 'Contrast Scale' for the MAPS micro-control thresholds. This is the one parameter which is strongly dependent on scene content and overall image intensity statistics. An intuitive sense of approximate range for a given image type and desired compression should emerge with experience. Initially, however, empirical exploration appears to be required here.

5.3 MAPS Decompression - DMAPS

This module generates the tonal decompressed image from the MAPS data stream. It also creates a companion MAPS 'resolution' image from this same stream but it is an ancillary product and will be illustrated
later. MAPS decompression is automatic and requires no interaction other than initial invocation of the module.

5.4 Subframe to Raster Conversion - RASTER

The only option to be specified in this module is the selection of the MAPS product type. Two products are relevant for the basic application of MAPS. The first is simply the re-establishment of the source image in an eight bit version and raster format with only the selected number of lines and pixels retained. Here, the user selects conversion of file IMAGE.DAT to file IRAST.DAT.

The other, more interesting, basic product is the MAPS decompressed tonal image. Here, the user selects conversion of file DMAPS.DAT to DRAST.DAT.

5.5 User Interaction Protocol ('Quick' Mode)

The following sequence of modules was invoked to illustrate the basic application of MAPS to image coding and reconstruction:

RUN SF  Conversion of source image to subframes
RUN MP  MAPS compression
RUN DM  MAPS decompression
RUN RS  Conversion of source image to raster
RUN RS  Conversion of MAPS image to raster
RUN AI  Assembly of images for printer display

The input image was chosen as the GIRL6.DAT 'toy' image and an 8x8 staggered subframe partition was used.
The SUBFRM protocol follows and is just as it evolved on the DEC-writer. User responses are left-justified along the edge of the listing. The listing is essentially self-descriptive:

```
RUN DR1:[50,27]SF

******************************************************************************
* MAPS RASTER TO SUBFRAME CONVERSION MODULE *
******************************************************************************

SOURCE IDENTIFICATION:

SOURCE RASTER FILENAME? (UP TO 9 CHARACTERS) FOR002 GIRL6

USER IMAGE NAME? (UP TO 8 CHARACTERS)
GIRL

SOURCE IMAGE POSITION:

NUMBER OF LINES TO SKIP? 0 (/ = NO CHNG)
/  NUMBER OF PIXELS TO SKIP? (< 4000) 0 (/ = NO CHNG)
/

SOURCE IMAGE SIZE:

NUMBER OF LINES TO PROCESS? 480 (/ = NO CHNG)
120
NUMBER OF PIXELS TO PROCESS? (UP TO 4000) 624 (/ = NO CHNG)
128
NUMBER OF BITS/PIXEL? (6 8) 8 (/ = NO CHNG)
6

SOURCE IMAGE PARTITION:

SUBFRAME EDGE? (8 16 32) 8 (/ = NO CHNG)
/  STAGGER GRID? (Y OR N) N

USER SPECIFICATION COMPLETE:
******************************************************************************

REVIEW? (Y OR N) N

CONVERTING IMAGE GIRL TO 254 SUBFRAMES
```

II-31
The MAPS protocol shows selection of the 'Quick' mode and the subsequent choice of a 'Contrast Scale' of 72. The 'SAVE ...' query is associated with retention of parameters for later use in the 'User' mode; it will be discussed in Section Eight. The resulting DEC-writer listing is:

```
RUN DR11[50,27]MP

******************************************************************************
* * MAPS COMPRESSION MODULE *
* *
******************************************************************************

USER OPTION MODES:

Q - QUICK MODE (SELECT CONTRAST SCALE ONLY)
U - USER PRE-DEFINED PARAMETERS FROM FILE MSET.DAT
F - FULL OPTION REVIEW AND SELECTIVE REVISION

MODE? (Q U F) Q
Q
CONTRAST SCALE? 20.0 (/ = NO CHNG) 72

USER SPECIFICATION COMPLETE:
******************************************************************************

REVIEW? (Y OR N) N
N
SAVE THESE PARAMETERS FOR FUTURE USE? (Y OR N) N
N
MAPS COMPRESSING IMAGE GIRL 120 LINES BY 128 PIXELS
MAPS FILE CONTAINS 7 512-BYTE RECORDS PLUS 413 BYTES IN THE LAST
MAPSEL DISTRIBUTION:

LEVEL: 0 1 2 3
COUNT: 1172 1571 422 32

OPTIMAL BIAS: - -6 -2 -0
+ 9 5 0

COMPRESSION RATIO: 2.882 : 1
BITS/PIXEL: 2.08177
MEAN SQUARE ERROR: 0.17221 %
```

Note that after completion of the compression task, module #2 returns a brief summary of results. These include: the number of 512-byte records required for the MAPS stream; the distribution of 'MAPSels' by size; optimal bias values for the subsequent pattern decompression (see Section Six); the compression level; and an overall fidelity measure in the form of the mean square error (MSE) in percent.
The protocol for DMAPS is very short:

```
RUN DR1:50,273DM

******************************************************************************
* * MAPS DECOMPRESSION/RESOLUTION IMAGE MODULE *
* *******************************************************************************

NO USER INPUTS REQUIRED

MAPS DECOMPRESSING IMAGE GIRL, 120 LINES BY 128 PIGELETS
>
```

The protocols for the two subframe-to-raster conversion runs are also self-explanatory. Note that RASTER reports both the user image name and the type of product being formed:

```
RUN DR1:50,273RS

******************************************************************************
* * MAPS SUBFRAME TO RASTER CONVERSION MODULE *
* *******************************************************************************

MAPS PRODUCT IMAGE TYPE:
I - IMAGE (ORIGINAL SOURCE)
D - DMAPS (MAPS DECOMPRESSED)
L - LEVEL (MAPS RESOLUTION CODES)
A - ADAPT (ADAPTIVELY SMOOTHED)
E - ERROR (DIFFERENCE)

TYPE? (I D L A E) DMAPS]
I

USER SPECIFICATION COMPLETE!
******************************************************************************

REVIEW? (Y OR N) N

CONVERTING IMAGE GIRL, FILE TYPE: IMAGE
TO 120 LINE BY 128 PIXEL RASTER, FILE TYPE: IRAST
>
```
RUN DR1:[50,27]RS

*****************************************************************************
* * MAPS SUBFRAME TO RASTER CONVERSION MODULE *
* * *****************************************************************************

MAPS PRODUCT IMAGE TYPE:

I - IMAGE (ORIGINAL SOURCE)
D - DMAPS (MAPS DECOMPRESSED)
L - LEVEL (MAPS RESOLUTION CODES)
A - ADAPT (ADAPTIVELY SMOOTHED)
E - ERROR (DIFFERENCE)

TYPE? (I D L A E) D[MAPS]

USER SPECIFICATION COMPLETE:
*****************************************************************************

REVIEW? (Y OR N) N

CONVERTING IMAGE GIRL , FILE TYPE: DMAPS
TO 120 LINE BY 128 PIXEL RASTER, FILE TYPE: DRAST

Finally, the protocol for ANNOTE image assembly follows:

RUN DR1:[50,27]AI

*****************************************************************************
* * IMAGE ASSEMBLY AND ANNOTATION MODULE *
* * *****************************************************************************

OUTPUT IMAGE SPECIFICATION:

OUTPUT FILE MODE:

R - GRAY SCALE RASTER IMAGE FILE "ANIMG.DAT"
P - LINE PRINTER PSEUDO IMAGE FILE "APRINT.DAT"

MODE? (R P) R

NUMBER OF LINES? 800 (/ = NO CHNG)
241
NUMBER OF PIXELS? (UP TO 128) 128 (/ = NO CHNG)
/COMPLEMENT BACKGROUND? (Y OR N) N
EMBEDDED IMAGES:

NUMBER OF IMAGES? (0 1 2) 0 (/ = NO CHNG)

IMAGE 1:

FILENAME? (UP TO 9 CHARACTERS) FOR002

IRAST

SKIP LINES INTO INPUT IMAGE? 0 (/ = NO CHNG)

/ SKIP PIXELS INTO INPUT IMAGE? (<4000) 0 (/ = NO CHNG)

NUMBER OF LINES? (UP TO 241) 241 (/ = NO CHNG)

120

NUMBER OF PIXELS? (UP TO 128) 128 (/ = NO CHNG)

/ STARTING LINE? (RANGE 1 - 122) 1 (/ = NO CHNG)

/ STARTING PIXEL? (RANGE 1 - 1) 1 (/ = NO CHNG)

/ COMPLEMENT IMAGE? (Y OR N) N

IMAGE 2:

FILENAME? (UP TO 9 CHARACTERS) FOR003

DRAST

SKIP LINES INTO INPUT IMAGE? 0 (/ = NO CHNG)

/ SKIP PIXELS INTO INPUT IMAGE? (<4000) 0 (/ = NO CHNG)

NUMBER OF LINES? (UP TO 241) 241 (/ = NO CHNG)

120

NUMBER OF PIXELS? (UP TO 128) 128 (/ = NO CHNG)

/ STARTING LINE? (RANGE 1 - 122) 1 (/ = NO CHNG)

122

STARTING PIXEL? (RANGE 1 - 1) 1 (/ = NO CHNG)

/ COMPLEMENT IMAGE? (Y OR N) N

EMBEDDED ANNOTATION:

NUMBER OF MESSAGES? (0 - 20) 0 (/ = NO CHNG)

/ USER SPECIFICATION COMPLETE:

*****************************************************************************

REVIEW? (Y OR N) N

ASSEMBLING AND ANNOTATING IMAGE:

> 241 LINES BY 120 PIXELS TO FILE "APRINT.DAT"

The resulting pseudo-image is displayed in Figure 5-1.
Figure 5-1. Basic MAPS Compression Example.  
Top - original; Bottom - MAPS Decompression
SECTION SIX

MAPS CONCEPTS AND PROCESSES

This section provides a synopsis of the key MAPS processes and concepts. More detailed discussions can be found in the references listed in Section 1.3

6.1 Image Partitioning

The MAPS partition of an image from the full frame down to the level of individual pixels is a two-stage process. First the frame is divided into square subframes, all of the same size. The pixel count for the subframe edge is required to be a power of two. Subframe sizes of 8x8 pixels, 16x16 pixels, or 32x32 pixels are allowed.

The subframes tesselate the image in either a square grid or a grid which is 'staggered' in one direction to give a 'brick wall' effect. Stagger is allowed only with the 8x8 subframe size and is intended to break up the perceptible 'blockiness' of the grid.

Within each subframe, the image is further divided by successive 'quartering'. This results in a series of nested 'quads', each quad having an edge pixel count which is a power of two. This division continues until the original pixel size is reached.

MAPS recodes the image from many simple fixed-sized pixels into a variable resolution pattern based on the image content. Each MAPS element or 'MAPSel' coincides with one of the natural quad units. MAPSels can range in size from original pixels up to entire subframes. The MAPSels are constrained, however, to give a complete (non-overlapping) tesselation of the image.
Thus, the MAPS partition processes involve a 'gridding' operation to form subframes, and a 'quadtree' division within the subframes. These concepts and associated labeling conventions are summarized in Figure 6-1.

6.2 Sequence Conventions

MAPS coding involves order conventions for sequencing the subframes and for sequencing elements within the subframes which allows element position information to remain implicit. That is, the element position is given by the location of the element in the storage sequence.

Square subframes are ordered in a simple coarse raster proceeding through rows of subframes in the 'pixel' direction and then advancing from row to row in the 'line' direction. Staggered subframes are ordered first by their startline and then by their position in the pixel direction. Thus, every eighth subframe along the pixel direction is given in sequence, and the startline is then advanced to the next row of every eighth subframe (see the stagger pattern in Figure 6-1).

Within each subframe, the nested quad pattern is traced from the lowest composite index to the highest. This results in the 'zig-zag' pattern through the source image as shown in the upper portion of Figure 6-2. This zig-zag pattern is also applicable to a valid sequence of MAPSels if the subpatterns within each larger element are collapsed to a point. An example of the resulting MAPS sequence is displayed in the bottom portion of Figure 6-2.

In essence, the MAPS order convention is that of a 'sequential quadtree'.

II-38
SUBFRAME STAGGER

Pixel Index Direction

\[
\begin{array}{c|c}
I_0 & I_1 \\
I_2 & I_3 \\
\end{array}
\]

Line Index Direction

\[
\begin{array}{c|c|c|c|c|c}
0_0 & 0_1 & 0_0 & 1_1 & 0_0 & 1_0 \\
2_0 & 3_0 & 2_0 & 3_0 & 2_0 & 3_0 \\
0_0 & 1_0 & 0_0 & 1_0 & 0_0 & 1_0 \\
2_0 & 3_0 & 2_0 & 3_0 & 2_0 & 3_0 \\
0_0 & 1_0 & 0_0 & 1_0 & 0_0 & 1_0 \\
2_0 & 3_0 & 2_0 & 3_0 & 2_0 & 3_0 \\
0_0 & 1_0 & 0_0 & 1_0 & 0_0 & 1_0 \\
2_0 & 3_0 & 2_0 & 3_0 & 2_0 & 3_0 \\
0_0 & 1_0 & 0_0 & 1_0 & 0_0 & 1_0 \\
2_0 & 3_0 & 2_0 & 3_0 & 2_0 & 3_0 \\
\end{array}
\]

Local Position

Resolution (Level)

QUAD LABELING

\[
I^L_L = \text{Gray Scale Position Level}
\]

QUAD NESTING

Figure 6-1. MAPS Image Partition Concepts
Figure 6-2. MAPS Sequence Concepts
6.3 Micro-fidelity Control

Formation of MAPSels larger than the original pixels involves successive evaluation of quad 'contrast'; testing of these contrasts against control thresholds; and composition of a quad into the next larger single element if no thresholds are exceeded. For each 'level' in the nest of quads, four contrasts are defined among the quad components as shown in the upper portion of Figure 6-3. A separate threshold is applied against each of these contrast types. This allows control on both adjacent intensity steps among the four elements and on the overall intensity range within the quad. The thresholds depend not only on the particular contrast type, but also on the level (element size) of the quad. This leads to a matrix of contrast control thresholds as depicted in the lower half of Figure 6-3.

In general, the threshold should be smaller for the 'step' contrasts than for the 'extreme' contrast. Moreover, the 'middle' step threshold may be set smaller than the 'outer' steps to preserve patterns where faint horizontal or vertical edges coincide with the quad centerlines.

Finally, the thresholds should decrease rapidly with increasing element size since small intensity differences are much more noticeable among larger blocks. These observations are summarized in the plot given as the upper portion of Figure 6-4.

The contrast threshold matrix may be set directly by selecting each of its components. However, the threshold loci suggested in Figure 6-4 can be generated from a smaller set of parameters. Specification from a four-parameter set is illustrated in the lower part of Figure 6-4. The first parameter is an overall 'contrast scale' which is the 'extreme' threshold for the level transition from 1x1 to 2x2 elements. The 'recursive taper base' provides exponential threshold decay with increasing level. The 'step fraction' specifies the 'middle step' in
SORTED INTENSITIES

GRAY-SCALE PATTERN (EXAMPLE)

HIGH
11 13 10 12
UPPER STEP CONTRAST
MIDDLE STEP CONTRAST
LOWER STEP CONTRAST
EXTREME CONTRAST

LOW

CONTRAST DEFINITIONS

CONTRASTS
EXTREME
MIDDLE STEP
LOWER STEP
UPPER STEP

LEVEL TRANSITIONS

0→1
1→2
...
(L-1)→L
...
(M-1)→M

THRESHOLD VALUES

CONTRAST CONTROL MATRIX

Figure 6-3. MAPS Contrast Control
CONTRAST THRESHOLD LOCi

EXTREME \((L \rightarrow L+1) = \text{EXTREME } (L-1 \rightarrow L) / B\)

MIDDLE \((L \rightarrow L+1) = F \times \text{EXTREME } (L \rightarrow L+1)\)

LOWER \((L \rightarrow L+1) = (F + \Delta) \times \text{EXTREME } (L \rightarrow L+1)\)

UPPER \((L \rightarrow L+1) = (F + \Delta) \times \text{EXTREME } (L \rightarrow L+1)\)

WHERE USER INPUTS SPECIFY:

- \(\text{EXTREME} (0 \rightarrow 1)\) - CONTRAST SCALE
- \(B\) - RECURSIVE TAPER BASE
- \(F\) - STEP FRACTION
- \(\Delta\) - STEP BIAS

PARAMETRIC MATRIX DEFINITION

Figure 6-4. MAPS Threshold Selection
terms of the 'extreme' threshold at a given level. Finally, the 'step bias' makes the 'outer' step thresholds larger than that for the 'middle step' at the same level.

Extensive empirical studies have shown that 'universal' values can be chosen for the taper base, step fraction, and step bias with little loss of performance over a wide range of imagery. These values are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taper Base, B</td>
<td>3.0</td>
</tr>
<tr>
<td>Step Fraction, F</td>
<td>0.5</td>
</tr>
<tr>
<td>Step Bias, Δ</td>
<td>0.1</td>
</tr>
</tbody>
</table>

The contrast scale, however, is strongly dependent on image content and intensity distribution. Thus, this is the one control parameter which must be chosen in the 'Quick' mode of MAPS option selection.

6.4 MAPSel Coding

MAPSels are increased in size until some quad threshold is exceeded, or until not all four sub-components of the quad are available (due to a prior threshold violation). The resulting sequence of MAPSels must then be coded in such a way that the image scene can be reconstructed.

As has already been discussed, position information remains implicit in the MAPSel stream. However, intensity and resolution information are given explicitly. For 8x8 subframes, there are only four allowed MAPSel levels so the resolution code occupies two bits per MAPSel. For compatibility with typical machine environments, such MAPSels are taken in groups of four and the four two-bit resolution codes are packed into a single byte. This is then followed by four bytes of intensity information.
For 16x16 and 32x32 subframes, there are five and six possible states, respectively. In this case, the MAPSels are taken in groups of three. The 16x16 subframe case requires 5x5x5 = 125 states to describe the resolution code triplet. The 32x32 case requires 6x6x6 = 216 states. Both of these fit within a single byte. Indeed, there is a bit 'left over' in the 16x16 case and this can be used for internal parity if desired. Again, the resolution code byte is followed by the corresponding intensity bytes (three).

Two forms of intensity coding are used. In the 'block' mode, a uniform intensity over the entire MAPSel is coded as an eight-bit byte. In the 'pattern' mode, only the top six bits of each byte contain direct intensity information. The two lowest-order intensity bits are replaced by a two-bit pattern which reflects one of four generic subpatterns for the quad. The relevant subpattern assignments are shown in Figure 6-5. The pattern bits are an automatic by-product of the contrast formation step so the compression computation is not significantly complicated by this process.

On decompression, the truncated intensity values are modified by a pattern of bias values which reflect the generic patterns. The bias values are constant over the image but vary among the MAPSel levels. Optimum image-wide biases (in a mean square error sense) can be determined by accumulating simple statistics during compression.

The pattern mode clearly makes no sense for MAPSels at level zero (1x1) since no pattern information is available. Also, the pattern mode is not very effective for large MAPSels since the loss due to intensity truncation from eight to six bits exceeds the size of the optimal biases in most cases. For middle-sized MAPSels, however, the improvement is dramatic. Thus, the user is given the option of selecting which levels will be coded in the 'block' mode and which will use the 'pattern' mode.
Figure 6-5. MAPS Pattern Mode
6.5 Adaptive 'Convolution'

The pattern mode makes dramatic improvements in MAPS image quality. Also, the use of subframe stagger tends to reduce the perception of 'blockiness' in the reconstructed imagery. However, such 'blockiness' can be masked much more effectively through use of a MAPS-based adaptive smoothing process. Note that this adaptive 'convolution' can be used with the pattern mode but is inconsistent with subframe stagger.

The adaptive smoothing is based on the following general observations. First, the size of a MAPSel is a rough estimate for the local 'correlation' length in the image. Thus, a local 'convolution' window of comparable size is appropriate for image smoothing. Second, the makeup of the 'surround' of the MAPSel to be smoothed contains useful control information for the convolution. Surrounding MAPSels which are much smaller than the target MAPSel give a priori indication of localized image activity which should not be included in the smoothing. Thus, 'surround' elements are activated only if they are no more than one level smaller than the target MAPSel. Finally, this restriction on activation means that the convolution depends on only sixteen regions - four in the (patterned) target MAPSel and twelve in the surround.

The geometry and numbering conventions for the adaptive 'convolution' are shown in Figure 6-6. The dynamic window size is adjusted to be one pixel narrower than the target MAPSel (to make it symmetric about the target pixel). Thus, any target pixel smoothing will depend on at most nine different local and surround elements. Moreover, the window weights can be pre-summed over each of these nine regions for each of the target pixel locations. The convolution then becomes a table-driven process involving just nine regional intensities, nine pre-summed weights, and nine activation flags for each target pixel.
Figure 6-6. MAPS Adaptive 'Convolution'
Since the weights are pre-summed, complex weighting functions can be used at no significant extra expense in computation. The user is given the choice between simple uniform weighting and a two-dimensional Gaussian weight with selectable spread.
SECTION SEVEN

MAPS USER OPTIONS

In this section, all of the user options for modules #1 through #6 are collected and tabulated. Where the role of the option is not clear from the context, a brief discussion is included. The first level of refinement of the total TransMAPS option space is reviewed in the following tabulation:

Option Groupings:

<table>
<thead>
<tr>
<th>Raster to Subframe Conversion:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Image Identification</td>
</tr>
<tr>
<td>Source Image Position Specification</td>
</tr>
<tr>
<td>Source Image Size Specification</td>
</tr>
<tr>
<td>Source Image Partition Specification</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAPS Compression:</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Mode Selection</td>
</tr>
<tr>
<td>Macro-Fidelity Control</td>
</tr>
<tr>
<td>Micro-Fidelity Control</td>
</tr>
<tr>
<td>Gray-Scale Manipulations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAPS Decompression and Resolution Image Formation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(User Transparent)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAPS Adaptive Image Smoothing:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution Weighting Specification</td>
</tr>
<tr>
<td>Dither Amplitude Specification</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAPS Difference Image Formation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Image Pair Selection</td>
</tr>
<tr>
<td>Difference Image Control</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subframe to Raster Conversion:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Product Image Type Selection</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image Assembly and Annotation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Image Specification</td>
</tr>
<tr>
<td>Embedded Input Image Specifications</td>
</tr>
<tr>
<td>Embedded Annotation Specifications</td>
</tr>
</tbody>
</table>
7.1 Raster to Subframe Conversion

The options in program SUBFRM are as follows:

Source Image Identification:
SOURCE RASTER FILENAME (Up to 9 characters)
USER IMAGE NAME (Up to 8 characters)

Source Image Position Specification:
SKIP LINES (Default 0)
SKIP PIXELS (Default 0)

Source Image Size Specification:
LINES
PIXELS (Skip pixels + Retained Pixels < 4001)
BITS/PIXEL (6 or 8, Default = 8)

Source Image Partition Specification:
SUBFRAME EDGE (8, 16, or 32, Default = 8)
SUBFRAME GRID (Square or Stagger, Default = Square)

Constraints: Stagger with 8x8 only,
Stagger incompatible with Adaptive Smoothing.

7.2 MAPS Compression

The MAPS Compression module has a particularly extensive option space. It is subdivided into three option classes - macro-fidelity control, micro-fidelity control, and gray-scale manipulations. The entire interaction is preceded by a 'user mode selection' which determines the overall interaction strategy.
Because of the complexity of the MAPS compression interaction, a separate 'road map' of the decision hierarchy has been included here as Figure 7-1. This presentation shows both the forward penetration of the hierarchy and the structure of return paths when local 'editing' modes are invoked.

7.2.1 User Interaction Mode

Three interaction strategies are available in the MAPS module - the 'Quick' mode, the 'User' mode, and the 'Full' mode. In the 'Quick' mode, only the overall image 'contrast scale' need be selected. This is the normal application of TransMAPS as a 'black box' image coding system.

The 'User' mode allows input of a set of user options defined on a previous interactive session with the MAPS module. The parameter set is stored on file MSET.DAT and overrides the default settings if this mode is selected. Note that the user can establish a new version of MSET.DAT for future use as the last interaction in the current run.

The 'Full' mode allows complete hierarchical penetration of the MAPS option space under user control (see Figure 7-1). Also, a decision to 'Review the Complete Specification' automatically returns the interaction to the beginning of the 'Full' mode sequence. Thus, the user can invoke a previous set-up with the 'User' mode and then edit this further with a 'Review' selection. This is helpful, for example, where a complex macro-fidelity partition is to be held fixed as various micro-control strategies are explored.
MAPS Compression User Option Decision Hierarchy (Subroutine USERM):

Mode Select: (Quick User Full)

'Quick' Mode
- Contrast Scale, Group 1

'User' Mode
- Input from File MSET.DAT

'Full' Mode

Macro-Fidelity Control: Review/Revise?
- (Display Current Image Partition)
  Matrix Row to Change?
  - --- Revise Selected Row

Micro-Fidelity Control: Review/Revise?
- (Loop on Active Groups)
  (Display Contrast Threshold Parameters, Group k)
  (Display Contrast Threshold Matrix, Group k)
  Revise Specifications? (No Scale-only Parameters Matrix)
  Scale-only:
  - --- Revise Contrast Scale, Group k
  Parameters:
  - Revise Contrast Scale, Group k
  - Revise Taper, Group k
  Revise Step Fraction, Group k
  - --- Revise Step Bias, Group k
  Matrix:
  - --- Revise Matrix Row, Group k
  - --- (Display Contrast Threshold Matrix, Group k)
  (End Loop)
  (Display Block/Pattern Assignment Vector)
  Revise Block/Pattern Vector

Gray-Scale manipulations: Review/revise?
- Code Space to Contrast Space Remapping:
  (Display Contrast Remap Breakpoint Pairs)
  Revise Contrast Remap?
  - Loop on Breakpoint Pairs
  Revise Next Breakpoint Pair
  (Exit from Loop if Code Space Exhausted)
  (End Loop)
- Code Space to Intensity Space Remapping:
  (Display Intensity Remap Breakpoint Pairs)
  Revise Intensity Remap?
  - Loop on Breakpoint Pairs
  Revise Next Breakpoint Pair
  (Exit from Loop if Code Space Exhausted)
  (End Loop)
  (Display Intensity Reset Node)
  Revise Intensity Reset: (M P L S T M)

Review Complete Specification?
- NO
- Retain Specification on File MSET.DAT?
- NO
- YES (Open and write MSET.DAT)
  To MAPS Compression Process

Figure 7-1. MAPS Compression Options Overview

II-53
7.2.2. Macro-Fidelity Control

Different micro-control strategies can be applied in different parts of the scene by establishing an appropriate macro-fidelity image partition. This involves specifying a 16x16 matrix which divides the source frame into 256 equal-area but distinct rectangular subpatches. The micro-fidelity control for each patch may be selected from any of up to four control parameter groups. Thus, the matrix component corresponding to each patch is assigned one of the digits from 1 to 4.

The normal default is to assign all patches to Group 1. However, the macro-fidelity image partition matrix can be changed on a row by row basis. Each row (16-element vector) update is followed immediately by display of the revised matrix. Thus, the patches can be reassigned until a satisfactory pattern for the scene content has been achieved.

Once the user has signalled an end to the macro-fidelity editing process, the matrix is scanned and all active groups (any subset of the numbers \{1,2,3,4\}) are noted. The interaction then automatically queries the user to set just those corresponding micro-fidelity controls.

7.2.3 Micro-Fidelity Control

For each active control group (up to four) there are three modes of control matrix specification available. The user may choose to specify only the 'contrast scale' for the group and leave the other parameters unchanged. Or, the user may elect the 'parametric' mode to change any among the 'contrast scale', 'taper base', 'step fraction', or 'step bias' for the group. A new matrix is then generated and displayed along with the parameter set.
Finally, the user may choose to edit the matrix components directly and by-pass the parametric generation. This approach is particularly appropriate where 'forced compositing' or 'forced resolution retention' is desired. Contrast thresholds which exceed the dynamic range of the possible contrast values will insure forced compositing; threshold values of 255 will work for 8-bit data. On the other hand, negative contrast thresholds will insure forced resolution retention. Examples are given in Sections Eight and Nine below.

The micro-fidelity control interaction also allows selection of the 'block' and 'pattern' mode assignments by MAPSel level. The choice is made by specifying an alphabetic string of B's and P's; the level association proceeds from left to right. Thus, BPFBPP would specify the 'pattern' mode for 2x2 and 4x4 MAPSels with the 'block' mode for the other levels. This particular 'block/pattern' vector is, in fact, the normal default selection.

7.2.4 Gray-Scale Manipulations

The options described in this section have the effect of allowing MAPS control to vary as a function of the image intensity. This is achieved in three ways - by contrast-space remapping, by intensity-space remapping, and by alternative intensity reset strategies.

The effect of a contrast threshold which varies with image intensity can be achieved by keeping a fixed threshold but using a non-linear mapping of the image data from code space (8 bits) to a new contrast space. Furthermore, if the contrast space is coded using more than eight bits, distinctness among the original levels can be preserved even though their relative spacing is changed. This capability is implemented in the MAPS module by allowing the user to specify a piecewise-linear mapping from 8-bit code space to 12-bit contrast space. The mapping can have up to
eight segments, continuous at segment boundaries but with a different slope for each segment.

The user specifies the mapping by responding with successive breakpoint coordinate pairs. That is, a segment end from code space and the corresponding point in contrast space are entered together as a two-element vector. The entry starts at the $(0,0)$ point and proceeds monotonically until the last entered pair exhausts code space $(255, < 4096)$. This approach allows the user to differentially retain certain features, such as radar strong returns, in very high fidelity while smoothing and compressing data from other portions of the gray-scale range.

A similar mapping from 8-bit code space to 12-bit intensity space allows non-linear formation of MAPSel intensities. This is appropriate, for example, if the image 8-bit code space represents a logarithmic encoding of the intensities. In this case, simple averaging of the code values in a quad is equivalent to taking the 'geometric' mean of the original signals rather than the 'arithmetic' mean. This would imply significant and systematic distortion of the radiometric information in the image.

Again, the MAPS module allows a piecewise-linear mapping from code space to intensity space; up to eight segments are also permitted here. This mapping could be used to approximate the transformation from code space back to the original intensity domain. Note that the module automatically forms the inverse demapping from intensity to code space in order to restore the final MAPSels to the proper range.

User specification of the intensity remapping proceeds in exactly the same manner as that for the contrast remapping. In both cases, the user must determine the basis to be used for the remappings before undertaking the interaction.
The final gray-scale manipulation option involves the selection of an intensity reset strategy. Here, the user is given six choices for the manner in which the quad is to be composited from its components. These choices are as follows:

- Mean of the four quad components;
- Pseudo-median of the quad (mean of the two middle elements);
- Lowest intensity in the quad;
- Second-lowest intensity in the quad;
- Third-lowest intensity in the quad;
- Highest intensity in the quad.

This choice may be used to avoid local drop-outs, isolated saturated points, or noise pulses of either sense. It is expected that this facility will be used only in very special image-dependent circumstances. The default selection is the simple mean which is the choice which minimizes the mean square error.

7.3 MAPS Decompression and Resolution Image Formation

As discussed in Section Five, this module proceeds automatically once it is invoked. However, two MAPS products are generated which may require some interpretation. The first is the DMAPS.DAT file which is the MAPS decompressed tonal image in subframe form. The only special feature here is that the optimal pattern biases are automatically applied to all levels for which the 'pattern' mode was selected. The bias information is transmitted as part of the standard MAPS file header.

The second product is the MAPS 'resolution' or 'level' image in file LEVEL.DAT. This is an 'image' formed by placing the host MAPSel resolution code in the upper three bits of each pixel. For 'pattern' mode MAPSels, the two pattern code bits are also included, shifted in two bits from the right edge of the byte. This product, then, gives a visual display of the MAPS resolution coding and is useful in understanding both
the MAPS process and the structure of a particular scene. Examples of
the MAPS level image are given in Sections Eight and Nine.

7.4 MAPS Adaptive Smoothing

Two option selections are required in the ADAPT module. They are:

Convolution Weighting:
   UNIFORM or GAUSSIAN   (Default Gaussian)
   SIGMA AT WINDOW CORNER  (Gaussian only, Default 2.0)

Dither Selection:
   DITHER AMPLITUDE   (Default 4.0)

A small dither may be added in the adaptive smoothing process to mask any
residual contouring. The amplitude is in gray levels relative to the
eight-bit code scale. The random variable is drawn from the system's
pseudo-random number generator.

7.5 MAPS Difference Image Formation

Three option selections are required in the DIFFER module. They are:

First Image of Difference Pair:
   IMAGE.DAT (source) or DMAPS.DAT (MAPS decompressed)

Second Image of Difference Pair:
   DMAPS.DAT or ADAPT.DAT (MAPS smoothed)

Difference Image Control:
   AMPLIFICATION FACTOR   (Default 10.0)
If DMAPS.DAT is chosen as the first image, the second automatically defaults to ADAPT.DAT. Thus, any of the three pairs - IMAGE-DMAPS, IMAGE-ADAPT, or DMAPS-ADAPT - can be formed.

The value of the amplification factor controls the type of difference image formed. A negative value results in a 'signed' difference with a neutral gray bias at gray value 127. A positive value results in an 'absolute' difference, amplified by the selected factor. A zero value results in the production of the fidelity statistics only, with no image file formed. The fidelity statistics are output to the printer listing file, EPRINT.DAT.

7.6 Subframe to Raster Conversion

The only option selection required in the RASTER module is selection of the desired product type. The range of possibilities is:

- **IMAGE.DAT** to **IRAST.DAT** Source image
- **DMAPS.DAT** to **DRAST.DAT** MAPS decompression
- **LEVEL.DAT** to **LRAST.DAT** MAPS resolution
- **ADAPT.DAT** to **ARAST.DAT** Adaptively smoothed
- **ERROR.DAT** to **ERAST.DAT** MAPS difference

The remaining two sections provide several examples of TransMAPS interactive protocols and sample results.
SECTION EIGHT

MAPS COMPRESSION: EXTENDED USE

This section presents examples of TransMAPS application with emphasis on MAPS compression options.

8.1 MAPS Compression Planning Form

Figure 8-1 presents a reduced photocopy of the MAPS Planning Form for the compression-phase tasks. The planning form summarizes the user option space for modules #1 (SUBFRM) and #2 (MAPS).

Note that the relevant file set for each module is shown. In addition, the three major option subgroups for the MAPS module - macro-fidelity control, micro-fidelity control, and gray-scale manipulations - are clearly distinguished. Finally, formal space is provided at the bottom of the form to record the output summary for the resultant MAPS run.

8.2 TransMAPS Compression Diagnostic Test Image

A special 'toy'-sized diagnostic image was developed as part of TransMAPS to test several aspects of the MAPS compression logic. This image is displayed in Figure 8-2 with an overlay of grid lines to show 'natural' MAPS boundaries relative to the various patterns. Four similar patterns are seen which differ only in scale. These structures contain generic quad geometries for MAPSels of sizes 1x1, 2x2, 4x4, and 8x8. In addition, one quarter of the structure for 16x16 MAPSels and single MAPSels of size 32x32 are represented.

For each of the four smallest MAPSel sizes, all possible generic quad patterns of the following types are included in the twelve quads of the
Figure 8-1. MAPS Compression Planning Form

II-61
Figure 8-2. MAPS Compression-Logic Diagnostic Image
basic structure:

- one dark MAPSel and three light MAPSels,
- two adjacent dark MAPSels and two adjacent light MAPSels, and
- three dark MAPSels and one light MAPSel.

At least one pattern of each of these types is also included for the 16x16 MAPSel size. Finally, the full frame contains examples of quads with four light MAPSels and others with four dark MAPSels for all five sizes - 1x1, 2x2, 4x4, 8x8, and 16x16.

This diagnostic image can then be used to implement all of the following compression-logic tests:

**Compression-Logic Diagnostic Tests:**

<table>
<thead>
<tr>
<th>Image Partition and Macro-Fidelity Control:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Image Line and Pixel Skips</td>
</tr>
<tr>
<td>Subframe Phasing (Square and Staggered)</td>
</tr>
<tr>
<td>Macro-Partition Group Assignment</td>
</tr>
</tbody>
</table>

**Micro-Fidelity Control:**

- Zigzag Sequencing
- Contrast Control as a Function of Transition Level
- Contrast Control as a Function of Contrast Type
- Pattern Code Assignment

**Gray-Scale Manipulation:**

- Contrast Space Quad Sort
- Intensity Space Quad Sort
- Intensity Reset Assignment

The diagnostic image is sufficiently small and regular so that the effects of input line and pixel skips can be predicted and verified directly. In addition, the effects of subframe phasing relative to the image structures and the natural MAPS boundaries can be varied by controlled line and pixel skips. These effects can also be predicted for both square and staggered subframes and can be verified very simply by displaying the MAPS resolution image.
Correct performance of the macro-fidelity image partition and corresponding control group assignment can also be established with this image. In this case, if only the first 128 lines of the 160 line by 128 pixel image are used, the macro-fidelity partition will divide the image into a 16x16 pattern of 8x8 pixel patches. Each of the complete scene structures can then be assigned to a different control group and treated with different micro-control strategies. Again, the results can be simply predicted and verified using the MAPS decompression and resolution image products.

Verification of the zig-zag sequencing is implicit in successful reconstruction of the varying MAPSel sizes following 'perfect fidelity' coding using zero contrast thresholds. Verification of contrast control as a function of transition level and contrast type is also possible using combinations of threshold which should yield varying 'forced composition' and 'forced resolution retention'. Finally, correct 'pattern code' assignments can be verified from the various 'two adjacent dark/two adjacent light' configurations which exhaust the generic 'pattern mode' geometries.

Selective combinations of thresholds to give controlled 'forced composition' plus 'forced resolution retention' can also be used to verify the quad sort results in both contrast and intensity space. The nearly exhaustive generic quad configurations also provide the vehicle for verifying the various intensity reset assignments.

Examples of several of these compression option explorations are contained in the protocol presented in the next subsection.

8.3 User Interaction Protocol ('User' and 'Full' Modes)

A completed MAPS-compression planning form for the example in this section is presented in Figure 8-3. The SUBFRM portion of the form
Figure 8-3. Planning Form for MTEST Example

II-65
reflects the usage of the diagnostic image, MTEST.DAT. It also indicates that only the first 128 lines (of 160) are to be used here. The image is to be partitioned into a square grid of 16x16 pixel subframes.

The MAPS portion of the planning form exhibits an extensive macro-fidelity-control image partition. The 'scene' has been segmented into four regions, each containing one of the full MAPS test structures. Group 1 is assigned to the patches covering the 1x1-MAPSel structure; group 2 to the 2x2-MAPSel structure; group 3 to the 4x4-MAPSel structure; and the remainder of the frame to group 4. This last group contains the entire 8x8 MAPSel test structure as well as isolated 16x16 and 32x32 MAPSels (see Figure 8-2).

This macro-fidelity partition allows different micro-control strategies to be applied to each of the full test structures. Direct specification of the contrast control matrix is used in each case. For group 1 (and the 1x1-MAPSel structure), two thresholds - the 'extreme' and 'upper step' for the level 0-1 transition - are set to the 'forced compositing' value, 255. All other group 1 thresholds are set to the 'perfect fidelity' value, 0. The effect of this specification should be to combine quads of 1x1 MAPSels which contain 'one dark and three light' components, and to leave the other quad types in the 1x1-MAPSel structure unchanged.

The contrast threshold matrix for group 2 (and the 2x2-MAPSel structure) is set to the 'perfect fidelity' condition except for the level 1-2 transition. This second row is set to the 'forced resolution retention' value, -1, for all four thresholds. Actually, any one of the thresholds set negative is sufficient. This means that the region controlled by group 2 will be coded by MAPSels no larger than 2x2, independent of the local scene content.
The contrast control matrix for group 3 (and the 4x4-MAPSel structure) is set with the 'perfect fidelity' condition in rows 1, 2, and 4. Row 3, corresponding to the level 2-3 transition, is set to the 'forced compositing' value, 255, in all four thresholds. The effect of this should be to composite all quads in the 4x4-MAPSel structure (level 2), to 8x8 MAPSels (level 3).

Finally, the remainder of the image, control group 4, is set completely to the 'perfect fidelity' value, 0. Thus, each element in region 4 should grow to its 'natural' MAPSel size.

Note that the 'block' mode is to be selected for all levels in this example.

Both the 'contrast remap' and 'intensity remap' are to be left in their default form - the identity mapping. However, 'pseudo-median' intensity reset is to be selected.

The results from the actual MAPS module run using these settings are entered at the bottom of the form. Photocopy reproductions of the actual interactive protocols from the resultant DEC-writer listings are displayed below.

The protocol for SUBFRM is as follows:

```
RUN DR1:[50,27]SF

# MAPS RASTER TO SUBFRAME CONVERSION MODULE

SOURCE IDENTIFICATION:
SOURCE RASTER FILENAME? (UP TO 9 CHARACTERS) FOR002
NTEST

USER IMAGE NAME? (UP TO 8 CHARACTERS)
NTEST 16

II-67
```
The protocol for MAPS is presented next. Actually, the macro-fidelity image partition matrix had been established on a previous run and saved on file MSET.DAT. (This sample file is also provided as part of the TransMAPS tape.) Thus, the 'User' mode was chosen to re-enter this data, and a 'Y' (yes) response to the 'REVIEW?' query was used to transfer back to the 'Full' mode for further editing. The MAPS protocol follows:

```
RUN DR1(50+27JMP

******************************************************************************
* * MAPS COMPRESSION MODULE * *
******************************************************************************

USER OPTION MODES:

Q - QUICK MODE (SELECT CONTRAST SCALE ONLY)
U - USER PRE-DEFINED PARAMETERS FROM FILE MSET.DAT
F - FULL OPTION REVIEW AND SELECTIVE REVISION

MODE? (Q U F) Q

U

USER SPECIFICATION COMPLETE:
******************************************************************************

REVIEW? (Y OR N) N

Y

II-68
```
MACRO-FIDELITY CONTROL: REVIEW/REVISE? (Y OR N) N

CURRENT IMAGE PARTITION

```
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 1
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 2
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 3
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 4
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 5
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 6
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 7
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 8
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 9
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 10
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 11
4 1 1 1 2 2 2 2 3 3 3 3 3 3 3 3 ROW 12
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 13
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 14
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 15
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ROW 16
```

ROW TO CHANGE? (1-16) (/= NO FURTHER CHNG)

MICRO-FIDELITY CONTROL: REVIEW/REVISE? (Y OR N) N

GROUP 1 CONTRAST THRESHOLD MATRIX

```
<table>
<thead>
<tr>
<th>E</th>
<th>M</th>
<th>L</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1-2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3-4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

ROW TO CHANGE? (1-4) (/= NO FURTHER CHNG)

SPECIFICATION MODES:
N - NO CHANGE
S - SCALE ONLY
P - PARAMETRIC
M - MATRIX

REVISE SPECIFICATIONS? (N S P M) N

MATRIX ROW TO CHANGE? (1-4) (/= NO FURTHER CHNG)

REVISE GROUP 1/LEVEL 0-1?

```
<table>
<thead>
<tr>
<th>E</th>
<th>M</th>
<th>L</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>255</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

GROUP 1 CONTRAST THRESHOLD MATRIX

```
<table>
<thead>
<tr>
<th>E</th>
<th>M</th>
<th>L</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>255</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1-2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3-4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

MATRIX ROW TO CHANGE? (1-4) (/= NO FURTHER CHNG)

II-69
GROUP 2 CONTRAST THRESHOLD MATRIX

<table>
<thead>
<tr>
<th>E</th>
<th>M</th>
<th>L</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1-2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3-4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

SPECIFICATION MODES:
N - NO CHANGE
S - SCALE ONLY
P - PARAMETRIC
M - MATRIX

REVISE SPECIFICATIONS? (N S P M) N

MATRIX ROW TO CHANGE? (1-4) (/ = NO FURTHER CHNG)

REVISE GROUP 2/LEVEL 1-2?

E | M | L | U |
|---|---|---|---|
| 0 | 0 | 0 | 0 | (/ = NO CHNG)

GROUP 2 CONTRAST THRESHOLD MATRIX

<table>
<thead>
<tr>
<th>E</th>
<th>M</th>
<th>L</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>2-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3-4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

MATRIX ROW TO CHANGE? (1-4) (/ = NO FURTHER CHNG)

GROUP 3 CONTRAST THRESHOLD MATRIX

<table>
<thead>
<tr>
<th>E</th>
<th>M</th>
<th>L</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1-2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3-4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

SPECIFICATION MODES:
N - NO CHANGE
S - SCALE ONLY
P - PARAMETRIC
M - MATRIX

REVISE SPECIFICATIONS? (N S P M) N

MATRIX ROW TO CHANGE? (1-4) (/ = NO FURTHER CHNG)

REVISE GROUP 3/LEVEL 2-3?

E | M | L | U |
|---|---|---|---|
| 0 | 0 | 0 | 0 | (/ = NO CHNG)

GROUP 3 CONTRAST THRESHOLD MATRIX

<table>
<thead>
<tr>
<th>E</th>
<th>M</th>
<th>L</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1-2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2-3</td>
<td>255</td>
<td>255</td>
<td>255</td>
</tr>
<tr>
<td>3-4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

MATRIX ROW TO CHANGE? (1-4) (/ = NO FURTHER CHNG)

II-70
GROUP 4 CONTRAST THRESHOLD MATRIX

<table>
<thead>
<tr>
<th>E</th>
<th>M</th>
<th>L</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1-2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3-4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

SPECIFICATION MODES:
- N - NO CHANGE
- S - SCALE ONLY
- P - PARAMETRIC
- M - MATRIX

REVISE SPECIFICATIONS? (N S P M) N

BLOCK/PATTERN ASSIGNMENT:
- LEVEL 01234
- MODE BBBDB

REVISE B/P VECTOR? BBBDB

GRAY-SCALE MANIPULATIONS: REVIEW/REVISE? (Y or N) N

CONTRAST SPACE REMAPPING: PIECEWISE LINEAR
(CODE SPACE/CONTRAST SPACE) BREAKPOINT PAIRS

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>255</td>
</tr>
</tbody>
</table>

REVISE CONTRAST REMAP? (Y or N) N

INTENSITY SPACE REMAPPING: PIECEWISE LINEAR
(CODE SPACE/INTENSITY SPACE) BREAKPOINT PAIRS

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>255</td>
</tr>
</tbody>
</table>

REVISE INTENSITY REMAP? (Y or N) N

INTENSITY RESET:
- M - MEAN OF QUAD
- P - PSEUDO-MEDIAN OF QUAD
- L - LOWEST IN QUAD
- S - SECOND IN QUAD
- T - THIRD IN QUAD
- H - HIGHEST IN QUAD

REVISE RESET? (M P L S T H) M

II-71
Note that the micro-fidelity control matrices had all been set to the 'perfect fidelity' condition as part of the prior MSET.DAT definition. Also, the 'block/pattern' vector had been set to all 'block' mode, BBBBBB. The only editing required was that to revise the contrast control matrices for each group and to update the intensity reset strategy.

The results for this example were retrieved and displayed by invoking modules DMAPS, RASTER (twice), and ANNOTE. The protocol for DMAPS is:
The RASTER runs converted both DMAPS.DAT and LEVEL.DAT. The RASTER protocols are:

```
RUN DR1:[50,27]DM

***************************************************************************
* * MAPS DECOMPRESSION/RESOLUTION IMAGE MODULE *
* * ***************************************************************************

NO USER INPUTS REQUIRED

MAPS DECOMPRESSING IMAGE MTEST16, 128 LINES BY 128 PIXELS
>

The RASTER runs converted both DMAPS.DAT and LEVEL.DAT. The RASTER protocols are:

```
RUN DR1:[50,27]RS

***************************************************************************
* * MAPS SUBFRAME TO RASTER CONVERSION MODULE *
* * ***************************************************************************

MAPS PRODUCT IMAGE TYPES:

I - IMAGE (ORIGINAL SOURCE)
D - DMAPS (MAPS DECOMPRESSED)
L - LEVEL (MAPS RESOLUTION CODES)
A - ADAPT (ADAPTIVELY SMOOTHEO)
E - ERROR (DIFFERENCE)

TYPE? (I D L A E) D[MAPS]

D

USER SPECIFICATION COMPLETE:
***************************************************************************

REVIEW? (Y OR N) N

N

CONVERTING IMAGE MTEST16, FILE TYPE: DMAPS

TO 128 LINE BY 128 PIXEL RASTER, FILE TYPE: DRAST
>

II-73
RUN DR1(S50,27)RS

******************************************************************************
* * MAPS SUBFRAME TO RASTER CONVERSION MODULE *
* *
******************************************************************************

MAPS PRODUCT IMAGE TYPE:

I - IMAGE (ORIGINAL SOURCE)
D - DMAPS (MAPS DECOMPRESSED)
L - LEVEL (MAPS RESOLUTION CODES)
A - ADAPT (ADAPTIVELY SMOOTHED)
E - ERROR (DIFFERENCE)

TYPE? (I D L A E) D [MAPS]
L

USER SPECIFICATION COMPLETE:
******************************************************************************

REVIEW? (Y OR N) N

CONVERTING IMAGE MTEST 16, FILE TYPE: LEVEL
TO 128 LINE BY 128 PIXEL RASTER; FILE TYPE: LRAST

Finally, the ANNOTE protocol is:

RUN DR1(S50,27)AI

******************************************************************************
* * IMAGE ASSEMBLY AND ANNOTATION MODULE *
* *
******************************************************************************

OUTPUT IMAGE SPECIFICATION:

OUTPUT FILE MODE:

R - GRAY SCALE RASTER IMAGE FILE "ANIMG.DAT"
P - LINE PRINTER PSEUDO IMAGE FILE "APRINT.DAT"

MODE? (R P) R
P

NUMBER OF LINES? 800 (/ = NO CHNG)
257

NUMBER OF PIXELS? (UP TO 128) 128 (/ = NO CHNG)
/

COMPLEMENT BACKGROUND? (Y OR N) N

II-74
EMBEDDED IMAGES:

> 

NUMBER OF IMAGES? (0 1 2) 0 (/ = NO CHNG)

2

IMAGE 1:

FILENAME? (UP TO 9 CHARACTERS) FOR002

/ DRAST

SKIP LINES INTO INPUT IMAGE? 0 (/ = NO CHNG)

/ SKIP PIXELS INTO INPUT IMAGE? (<=4000) 0 (/ = NO CHNG)

/ NUMBER OF LINES? (UP TO 257) 257 (/ = NO CHNG)

128

NUMBER OF PIXELS? (UP TO 128) 128 (/ = NO CHNG)

/ STARTING LINE? (RANGE 1 - 130) 1 (/ = NO CHNG)

/ STARTING PIXEL? (RANGE 1 - 1) 1 (/ = NO CHNG)

/ COMPLEMENT IMAGE? (Y OR N) N

N

IMAGE 2:

FILENAME? (UP TO 9 CHARACTERS) FOR003

/ LRAST

SKIP LINES INTO INPUT IMAGE? 0 (/ = NO CHNG)

/ SKIP PIXELS INTO INPUT IMAGE? (<=4000) 0 (/ = NO CHNG)

/ NUMBER OF LINES? (UP TO 257) 257 (/ = NO CHNG)

128

NUMBER OF PIXELS? (UP TO 128) 128 (/ = NO CHNG)

/ STARTING LINE? (RANGE 1 - 130) 1 (/ = NO CHNG)

/ STARTING PIXEL? (RANGE 1 - 1) 1 (/ = NO CHNG)

/ COMPLEMENT IMAGE? (Y OR N) N

N

EMBEDDED ANNOTATION:

NUMBER OF MESSAGES? (0 - 20) 0 (/ = NO CHNG)

/ USER SPECIFICATION COMPLETE:

REVIEW? (Y OR N) N

N

ASSEMBLING AND ANNOTATING IMAGE:

> 257 LINES BY 128 PIXELS TO FILE "APRINT.DAT"
The resultant MAPS 'decompressed' image and the corresponding MAPS 'level' image are displayed in Figure 8-4. For the 1x1-MAPSel structure, the 'one dark and three light' quads are seen to be forced to 2x2 form and to show up as 'light' MAPSels. This intensity reset is a consequence of the 'pseudo-median' selection and the fact that the two 'middle' intensities are both 'light'. The other quads of 1x1 MAPSels are left unchanged. Thus, the predictions for this structure are all borne out.

In the region corresponding to group 2, the resolution image shows that 2x2 MAPSels are used throughout. Again, the predictions are verified.

In the group 3 region, all 4x4 MAPSels are seen to be composited to 8x8 form. The 'one dark and three light' quads go to all 'light'. The 'two adjacent dark and two adjacent light' quads go to a 'mid-gray'. The 'three dark and one light' quads go to all 'dark'. Each of these is consistent with 'forced compositing' and 'pseudo-median reset' as expected.

Finally, region 4 shows the predicted 'perfect fidelity' decompression and 'natural MAPSel' resolution-image structure.

This example demonstrates the diagnostic and verification power of the special test image, MTEST.DAT. It also shows the potential of this image to illustrate the detailed performance of the MAPS compression options. The user is encouraged to exploit this image for further development of MAPS process understanding and 'intuition'.
Figure 8-4. MAPS Decompressed and Resolution Images for Diagnostic Test Example

11-77
SECTION NINE

MAPS PRODUCT GENERATION: EXTENDED USE

This section presents examples of TransMAPS application with emphasis on MAPS Product formation options.

9.1 MAPS Product Generation Planning Form

Figure 9-1 presents a reduced photocopy of the MAPS Planning Form for the product generation tasks. The planning form summarizes the user options for modules #3 (DMAPS), #4 (ADAPT), #5 (DIFFER) and #6 (RASTER).

These modules have relatively few interactive parameters compared with the compression-phase tasks. However, the file structure is much more extensive and complex (see Figure 2-1). The relevant files for each module are listed on the form.

9.2 User Interaction Protocol (All Modules)

The remaining subsections describe a series of examples in which the MAPS compression level is held fixed at essentially two bits per pixel. The source image for the series is the 'toy' (120 line x 128 pixel) GIRL6.DAT image. The examples depict the evolution of increasing MAPS image 'quality' under alternative compression and product generation strategies. Seven examples make up the series and the comparative control states are summarized in the following tabulation:
MAPS Planning Form
(Allowed Range) [Default]

Date

Image Name

Compression

MODULE

USER INPUTS

RUN DMAPS

FILES:
IN - MAPS.DAT
OUT - DMAPS.DAT
OUT - LEVEL.DAT

RUN ADAPT /NOT WITH STAGGER GRID/

FILES:
IN - DMAPS.DAT
IN - LEVEL.DAT
OUT - ADAPT.DAT

RUN DIFFER

FILES:
IN - IMAGE.DAT
IN - MAPS.DAT
IN - ADAPT.DAT
OUT - ERROR.DAT
OUT - EPRINT.DAT (LISTING)

RUN RASTER

FILES:
IN - IMAGE.DAT
IN - MAPS.DAT
IN - DMAPS.DAT
IN - LEVEL.DAT
IN - ADAPT.DAT
IN - ERROR.DAT
OUT - IRAST.DAT
OUT - DRAST.DAT
OUT - LRAST.DAT
OUT - ARAST.DAT
OUT - ERAST.DAT

Convolution Weighting (Gaussian Uniform) [6]
Sigma multiple at window corner [2.0] /Gaussian Only/
Random Dither Amplitude [0.0]

Difference Parameter:
(<0 Signed *0 Statistics Only >0 Amplification Factor)[10]

MAPS Raster Product (Image DMAPS Level ADAPT Error) [DMAPS]

Type matches Input

Figure 9-1. MAPS Product Generation Planning Form

II-79
Product Generation Examples:

<table>
<thead>
<tr>
<th>Example</th>
<th>Contrast Scale</th>
<th>Block/Pattern</th>
<th>Convolution</th>
<th>Dither</th>
</tr>
</thead>
<tbody>
<tr>
<td>2X2 MEAN</td>
<td>Force Level 1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DMAPS B2</td>
<td>72</td>
<td>BPPP</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ADAPT B2</td>
<td>72</td>
<td>BPPP</td>
<td>Uniform</td>
<td>0</td>
</tr>
<tr>
<td>DMAPS P2</td>
<td>72</td>
<td>BPPP</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ADAPT P2</td>
<td>72</td>
<td>BPPP</td>
<td>Uniform</td>
<td>0</td>
</tr>
<tr>
<td>GAUSS P2</td>
<td>72</td>
<td>BPPP</td>
<td>Gaussian;2</td>
<td>0</td>
</tr>
<tr>
<td>DITHER 8</td>
<td>72</td>
<td>BPPP</td>
<td>Gaussian;2</td>
<td>0</td>
</tr>
</tbody>
</table>

A sample protocol from one of the examples is exhibited as the remainder of this subsection. The protocol corresponds to the 'GAUSS P2' case and involves all seven modules in TransMAPS. The sequence of module invocations is as follows:

```
RUN SF
RUN MP
RUN DM
RUN AD
RUN DF
RUN RS (Convert ADAPT.DAT)
RUN RS (Convert ERROR.DAT)
RUN AI
```

By this point, the protocols should be self-explanatory so they are displayed sequentially without interruption:

```
RUN DR1:[50,27]SF
******************************************************************
* * MAPS RASTER TO SUBFRAME CONVERSION MODULE * *
* * ******************************************************************
SOURCE IDENTIFICATION:
SOURCE RASTER FILENAME? (UP TO 9 CHARACTERS) FOR002 GIRL6
USER IMAGE NAME? (UP TO 8 CHARACTERS)
GIRL 6x8
```
SOURCE IMAGE POSITION:

NUMBER OF LINES TO SKIP? 0 (/ = NO CHNG)

NUMBER OF PIXELS TO SKIP? (< 4000) 0 (/ = NO CHNG)

SOURCE IMAGE SIZE:

NUMBER OF LINES TO PROCESS? 480 (/ = NO CHNG)

NUMBER OF PIXELS TO PROCESS? (UP TO 4000) 624 (/ = NO CHNG)

NUMBER OF BITS/PIXEL? (6 8) 8 (/ = NO CHNG)

SOURCE IMAGE PARTITION:

SUBFRAME EDGE? (8 16 32) 8 (/ = NO CHNG)

STAGGER GRID? (Y OR N) N

USER SPECIFICATION COMPLETE:

REVIEW? (Y OR N) N

CONVERTING IMAGE GIRL 8x8 TO 240 SUBFRAMES

*****************************************************************
* MAPS COMPRESSION MODULE *
*****************************************************************

USER OPTION MODES:

Q - QUICK MODE (SELECT CONTRAST SCALE ONLY)
U - USER PRE-DEFINED PARAMETERS FROM FILE MSET.DAT
F - FULL OPTION REVIEW AND SELECTIVE REVISION

MODE? (Q U F) Q

MACRO-FIDELITY CONTROL: REVIEW/REVISE? (Y OR N) N

II-81
MICRO-FIDELITY CONTROL: REVIEW/REVISE? (Y OR N) N

GROUP 1 CONTRAST THRESHOLD PARAMETERS
CONTRAST SCALE  20.0
TAPER           3.0
STEP FRACTION   0.5
STEP BIAS       0.1

GROUP 1 CONTRAST THRESHOLD MATRIX
<table>
<thead>
<tr>
<th>E</th>
<th>M</th>
<th>L</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>20</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>1-2</td>
<td>7</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2-3</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

SPECIFICATION MODES:
N - NO CHANGE
S - SCALE ONLY
P - PARAMETRIC
M - MATRIX

REVISE SPECIFICATIONS? (N S P M) N

GROUP 1 CONTRAST SCALE? 20.0 (/ = NO CHNG)

GROUP 1 CONTRAST THRESHOLD PARAMETERS
CONTRAST SCALE  72.0
TAPER           3.0
STEP FRACTION   0.5
STEP BIAS       0.1

GROUP 1 CONTRAST THRESHOLD MATRIX
<table>
<thead>
<tr>
<th>E</th>
<th>M</th>
<th>L</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>72</td>
<td>36</td>
<td>43</td>
</tr>
<tr>
<td>1-2</td>
<td>24</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>2-3</td>
<td>8</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

SPECIFICATION MODES:
N - NO CHANGE
S - SCALE ONLY
P - PARAMETRIC
M - MATRIX

REVISE SPECIFICATIONS? (N S P M) N

B/P VECTOR ASSIGNMENT:
LEVEL 0123
MODE BPPB

REVISE B/P VECTOR? BPPB
BPPP

II-82
GRAY-SCALE MANIPULATIONS: REVIEW/REVISE? (Y OR N) N

USER SPECIFICATION COMPLETE:
*****************************************************

REVIEW? (Y OR N) N

SAVE THESE PARAMETERS FOR FUTURE USE? (Y OR N) N

MAPS COMpressing IMAGE GIRL 8x8, 120 LINES BY 128 PIXELS

MAPS FILE CONTAINS 7 512-BYTE RECORDS PLUS 234 BYTES IN THE LAST

MAPSEL DISTRIBUTION:

LEVEL: 0 1 2 3
COUNT: 1052 1557 425 20

OPTIMAL BIAS: -6 -2 0
+ 9 5 3

COMPRESSION RATIO: 3.017:1

BITS/PIXEL: 1.98854

MEAN SQUARE ERROR: 0.18080 %

RUN DR1:[50:27]DM

***************************************************************
* * MAPS DECOMPRESSION/RESOLUTION IMAGE MODULE *
* ***************************************************************

NO USER INPUTS REQUIRED

MAPS DECOMPRESSING IMAGE GIRL 8x8, 120 LINES BY 128 PIXELS

II-83
RUN DR1:[50,27]AD

***************************************************************************
* * MAPS ADAPTIVE SMOOTHING MODULE *
* *
***************************************************************************

CONVOLUTION WEIGHTING:

UNIFORM OR GAUSSIAN? (U G) G

SIGMA MULTIPLE AT WINDOW CORNER? 2.0 (/ = NO CHNG)

RANDOM DITHER:

AMPLITUDE? 4.0 (/ = NO CHNG)

USER SPECIFICATION COMPLETE:

***************************************************************************

REVIEW? (Y OR N) N

MAPS ADAPTIVE SMOOTHING IMAGE GIRL 8x8, 120 LINES BY 128 PIXELS

RUN DR1:[50,27]DF

***************************************************************************
* * MAPS DIFFERENCE IMAGE MODULE *
* *
***************************************************************************

INPUT IMAGE TYPES:

I - IMAGE (ORIGINAL SOURCE) IMAGE1 ONLY
D - DMAPS (MAPS DECOMPRESSED)
A - ADAPT (ADAPTIVELY SMOOTHED) IMAGE2 ONLY

IMAGE1? (I D) I [IMAGE]

IMAGE2? (D A) D [MAPS]

DIFFERENCE IMAGE TYPE:

AMPLIFICATION FACTOR
<0 SIGNED AND BIAS DIFFERENCE IMAGE
=0 NO DIFFERENCE IMAGE, STATISTICS ONLY
>0 AMPLIFIED DIFFERENCE IMAGE
(VALUE IS AMPLIFICATION)

FACTOR? 10. (/ = NO CHNG)

USER SPECIFICATION COMPLETE:

***************************************************************************

REVIEW? (Y OR N) N

DIFFERENCING GIRL 8x8 TYPE IMAGE, VS GIRL 8x8 TYPE ADAPT

II-84
RUN DR1:[50,27]RS

******************************************************************************
* * MAPS SUBFRAME TO RASTER CONVERSION MODULE *
******************************************************************************

MAPS PRODUCT IMAGE TYPE:

I - IMAGE (ORIGINAL SOURCE)
D - DMAPS (MAPS DECOMPRESSED)
L - LEVEL (MAPS RESOLUTION CODES)
A - ADAPT (ADAPTIVELY SMOOTHED)
E - ERROR (DIFFERENCE)

TYPE? (I D L A E) D[MAPS]


USER SPECIFICATION COMPLETE:
******************************************************************************

REVIEW? (Y OR N) N

CONVERTING IMAGE GIRL 8x8 FILE TYPE: ADAPT TO 120 LINE BY 128 PIXEL RASTER, FILE TYPE: ARAST

RUN DR1:[50,27]RS

******************************************************************************
* * MAPS SUBFRAME TO RASTER CONVERSION MODULE *
******************************************************************************

MAPS PRODUCT IMAGE TYPE:

I - IMAGE (ORIGINAL SOURCE)
D - DMAPS (MAPS DECOMPRESSED)
L - LEVEL (MAPS RESOLUTION CODES)
A - ADAPT (ADAPTIVELY SMOOTHED)
E - ERROR (DIFFERENCE)

TYPE? (I D L A E) D[MAPS]


USER SPECIFICATION COMPLETE:
******************************************************************************

REVIEW? (Y OR N) N

CONVERTING IMAGE GIRL 8x8 FILE TYPE: ERROR A-I TO 120 LINE BY 128 PIXEL RASTER, FILE TYPE: ERAST

>
RUN DR1:[50,27]AI

******************************************************************************
* IMAGE ASSEMBLY AND ANNOTATION MODULE *
******************************************************************************

OUTPUT IMAGE SPECIFICATION:

OUTPUT FILE MODE:

R - GRAY SCALE RASTER IMAGE FILE "ANIMG.DAT"
P - LINE PRINTER PSEUDO IMAGE FILE "APRINT.DAT"

MODE? (R P) R
P

NUMBER OF LINES? 800 (/ = NO CHNG)
256
NUMBER OF PIXELS? (UP TO 128) 128 (/ = NO CHNG)
/
COMPLEMENT BACKGROUND? (Y OR N) N

EMBEDDED IMAGES:

NUMBER OF IMAGES? (0 1 2) 0 (/ = NO CHNG)
2

IMAGE 1:

FILENAME? (UP TO 9 CHARACTERS) FORO02
ARAST

SKIP LINES INTO INPUT IMAGE? 0 (/ = NO CHNG)
/
SKIP PIXELS INTO INPUT IMAGE? (4000) 0 (/ = NO CHNG)
/
NUMBER OF LINES? (UP TO 256) 256 (/ = NO CHNG)
120
NUMBER OF PIXELS? (UP TO 128) 128 (/ = NO CHNG)
/
STARTING LINE? (RANGE 1 - 137) 1 (/ = NO CHNG)
/
STARTING PIXEL? (RANGE 1 - 1) 1 (/ = NO CHNG)
/
COMPLEMENT IMAGE? (Y OR N) N

IMAGE 2:

FILENAME? (UP TO 9 CHARACTERS) FORO03
ERAST

SKIP LINES INTO INPUT IMAGE? 0 (/ = NO CHNG)
/
SKIP PIXELS INTO INPUT IMAGE? (4000) 0 (/ = NO CHNG)
/
NUMBER OF LINES? (UP TO 256) 256 (/ = NO CHNG)
120
NUMBER OF PIXELS? (UP TO 128) 128 (/ = NO CHNG)
/
STARTING LINE? (RANGE 1 - 137) 1 (/ = NO CHNG)
137
STARTING PIXEL? (RANGE 1 - 1) 1 (/ = NO CHNG)
/
COMPLEMENT IMAGE? (Y OR N) N

II-86
EMBEDDED ANNOTATION:

NUMBER OF MESSAGES? (0 - 20) 0 (/ = NO CHNG)

MESSAGE 1!

ORIENTATION IN FRAME, TOP OF SYMBOL TOWARD:
T - TOP
B - BOTTOM
L - LEFT
R - RIGHT

ORIENTATION? (T B L R) T

MESSAGE 1 LENGTH
0 - 8 CHARACTERS AT 1X
0 - 4 CHARACTERS AT 2X
0 - 2 CHARACTERS AT 3X
0 - 2 CHARACTERS AT 4X

CHARACTER COUNT? 0 (/ = NO CHANGE, 0 = DELETE)

MESSAGE 1 SYMBOL SIZE? (1 ) 1 (/ = NO CHNG)

MESSAGE CENTER AT LINE? ( 8 - 248) 8 (/ = NO CHNG)

MESSAGE CENTER AT PIXEL? ( 64 - 64) 64 (/ = NO CHNG)

COMPLEMENT MESSAGE 1? (Y OR N) N

ALLOWED CHARACTERS

NUMERALS: 0 1 2 3 4 5 6 7 8 9
PUNCTUATION: ( ), ; , Space
ARITHMETIC: + - * / %
RELATIONAL: = < > f(<or=) $(>or=)
DIRECTIONAL: e (East)
(ARROWS) n (North)
w (West)
s (South)
a (NE Northeast) w + e
b (NW Northwest)
c (SW Southwest) c d
d (SE Southeast) s

MESSAGE 1 TEXT? ( 8 CHARACTERS)

-------------
GAUSS P2

MSG LINE PIX TEXT
1 T 1X 128 64 GAUSS P2

MESSAGE TO CHANGE? (1 - 1) (/ = NO FURTHER CHNG)

USER SPECIFICATION COMPLETE:
****************************

REVIEW? (Y OR N) N

ASSEMBLING AND ANNOTATING IMAGE:

256 LINES BY 128 PIXELS TO FILE *APRINT.DAT*

II-87
9.3 Resolution (level) Image

Figure 9-2 shows the source frame and the resolution or level image which exhibits the same form for all of the runs at contrast scale 72. The white areas correspond to 1x1 coding, the 'dots' to 2x2's, the 'minuses' to 4x4's and the 'pluses' to 8x8's. Note that MAPS concentrates its resources at points which define the key features of the image content.

Figure 9-3 presents the results of simple 2x2 averaging to achieve two bits per pixel. This then forms the baseline for comparison with MAPS products at the same compression level. This case was established using TransMAPS with 'forced compositing' (threshold 255) for the level 0-1 transition and 'forced resolution retention' (threshold -1) for the level 1-2 transition.

The lower half of Figure 9-3 portrays the results of the DIFFER Module applied between the source (1x1 throughout) and the 2x2 MEAN image. An amplification factor of 6.0 was used. The corresponding error histogram is displayed in Figure 9-4; this is the EPRINT.DAT file from DIFFER.

9.4 Block Decompression

Figure 9-5 shows the MAPS decompression and amplified difference image (factor = 6.0) for simple 'block' mode compression at all levels. The error histogram is shown in Figure 9-6. Note that MAPS definition around the eyes, mouth, and jaw line is considerably sharper than in the 2x2 MEAN case. However, the image is quite 'blocky' elsewhere.

9.5 Block Mode with Uniform Adaptive Smoothing

Figure 9-7 corresponds to subsequent adaptive smoothing of the 'block' mode results in Figure 9-5. The comparable error histogram is given in Figure 9-8. Note that the smoothing process does much to eliminate the perceptible artifacts. However, the smoothing process is relatively expensive in computation time.
9.6 Pattern Decompression

Figure 9-9 shows the MAPS decompression and difference image with the 'pattern' mode used throughout. The error histogram is given in Figure 9-10. This case shows significant sharpening of features relative to the 'block' mode and is quite acceptable even without further smoothing. The difference image exhibits a mostly 'salt and pepper' texture which is largely decorrelated from the scene content. Note that this is the normal default mode recommended for MAPS use and is very efficient computationally.

9.7 Pattern Mode with Uniform Adaptive Smoothing

Figure 9-11 corresponds to the adaptive smoothing of the 'pattern' image in Figure 9-9; the error histograms are shown in Figure 9-12. Here, 'uniform' convolution weighting was used in the smoothing process. This version is slightly better than the direct 'pattern' mode but at considerable extra computation.

9.8 Pattern Mode with Gaussian Adaptive Smoothing

Figure 9-13 depicts the same case as Figure 9-11 except that 'Gaussian' convolution weighting was substituted for 'uniform' weights. The spread of the weighting function was chosen so that the 2.0 sigma point occurs in the corner of the convolution window. Very small differences can be discerned between the results in Figures 9-11 and 9-13. The error histogram for the Gaussian case is presented in Figure 9-14.

9.9 Pattern/Gaussian Mode with Dither

Figure 9-15 repeats the Gaussian-smoothed results except that a random dither has been added to the 'smoothing' process. The maximum amplitude of the dither is 8 gray levels (of 256) with the actual sign and size for each pixel determined from a pseudo-random number generator. Figure 9-16 gives the matching error histogram. This example is seen to have less perceptible intensity 'contouring' than the cases without dither.
Figure 9-2. MAPS Resolution Image at 2 bits/pixel

II-90
Figure 9-3. Simple Averaging to Achieve 2 bits/pixel
Figure 9-4. Error Histogram For 2x2 MEAN Example
Figure 9-5. MAPS Block Mode at 2 bits/pixel

II-93
Figure 9-6. Error Histogram for DMAPS B2 Block Mode Example
Figure 9-7. MAPS Block Mode With Adaptive Smoothing (Uniform Weight)
Figure 9-8. Error Histogram for ADAPT B2 Block Mode Example
Figure 9-9. MAPS Pattern Mode at 2 bits/pixel
Figure 9-10. Error Histogram For DMAPS P2 Pattern Mode Example
Figure 9-11. MAPS Pattern Mode With Adaptive Smoothing (Uniform Weight)

II-99
DIFERENCE STATISTICS:

IMAGE1: GINU, TYPE: IMAGE, MITS/PIXEL: 6.00000
IMAGE2: GINU, TYPE: ADAPT, MITS/PIXEL: 1.90856
AMPLIFIED DIFFERENCE IMAGE PRODUCED: 240 SUBFRAMES, AMPLIFICATION: 6.0
DIFFERENCES, IMAGE1-IMAGE2: MEAN = 0.021, RMS 6.173, MSE 0.165997, RMSE 0.688028
IMAGE 1 (UNIT = 24 PIXELS)

Figure 9-12. Error Histogram for ADAPT P2 Pattern Mode Example
Figure 9-13. MAPS Pattern Mode With Adaptive Smoothing (Gaussian Weight)
Figure 9-14. Error Histogram for GAUSS P2 Pattern Mode Example
Figure 9-15. MAPS Pattern/Gaussian Mode With Dither
(Amplitude = 8)
9.10 Mean Square Error Performance Comparisons

The qualitative image improvement described for the above sequence is also borne out by the quantitative fidelity measures. The histograms in Figures 9-4, 9-6, 9-8, 9-10, 9-12, and 9-14 exhibit progressive narrowing with a slight relaxation when random dither is added (Figure 9-16). The following tabulation of summary error statistics reflects this improvement as well:

<table>
<thead>
<tr>
<th>Example</th>
<th>Mean Square Error</th>
<th>Relative MSE</th>
<th>RMS Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2X2 MEAN</td>
<td>0.66055 %</td>
<td>2.73859 %</td>
<td>12.317 %</td>
</tr>
<tr>
<td>DMAPS B2</td>
<td>0.30531 %</td>
<td>1.26581 %</td>
<td>8.374 %</td>
</tr>
<tr>
<td>ADAPT B2</td>
<td>0.27857 %</td>
<td>1.15492 %</td>
<td>7.998 %</td>
</tr>
<tr>
<td>DMAPS P2</td>
<td>0.18019 %</td>
<td>0.74706 %</td>
<td>6.433 %</td>
</tr>
<tr>
<td>ADAPT P2</td>
<td>0.16595 %</td>
<td>0.68802 %</td>
<td>6.173 %</td>
</tr>
<tr>
<td>GAUSS P2</td>
<td>0.16167 %</td>
<td>0.67027 %</td>
<td>6.093 %</td>
</tr>
<tr>
<td>DITHER B</td>
<td>0.17555 %</td>
<td>0.72780 %</td>
<td>6.349 %</td>
</tr>
</tbody>
</table>

From these summary statistics, it is seen that the 'block' mode is significantly better than simple averaging and the 'pattern' mode gives a further dramatic improvement. Although it is a heuristic technique, adaptive smoothing is also seen to yield a small but consistent enhancement in fidelity. Finally, the slight loss from the addition of dither seems more than compensated by its improvement of visual quality.

9.11 A 'Real'-Image Example

This closing subsection presents an example of the application of TransMAPS to a 'real' (video-sized) image - the BLDGIMG.DAT file which is supplied with the package. The interaction is summarized on the three planning forms displayed as Figures 9-17, 9-18, and 9-19. Not shown are two ANNOTE protocols, each of which assembles two image frames prior to the final four-frame composite. This is, however, an example of the recursive use of ANNOTE.
Figure 9-17. Compression Planning Form - Building Scene
Figure 9-18. Product Generation Planning Form - Building Scene

II-107
Figure 9-19. Annotation Planning Form - Building Composite
Figure 9-20 presents the overall visual results. The frame contains the original source image, the MAPS decompressed image (at 2 bits/pixel), the resolution or level image, and an amplified difference image which has been complemented in ANNOTE so large differences are light in a dark frame. Finally, the difference statistics for this example are displayed as Figure 9-21.
Figure 9-20. 'Real'-Image MAPS Example-Building Scene
Figure 9.21: Error Histogram for Building Scene
APPENDIX

FULL SIZE TransMAPS PLANNING FORMS

(Suitable for Photocopying)
RUN SUBRM

FILES:
- IN - USER RASTER
- OUT - IMAGE.DAT

SOURCE IMAGE FILE NAME (---)
USER IMAGE NAME (---)
SKIP LINES [0] (---)
SKIP PIXELS [0] (---)
PROCESS LINES (---)
PROCESS PIXELS (---)
BIT/SPIXEL (6 8) [8]
SUBFRAME SIZE (8 16 32) [8] SUBFRAME GRID (SQUARE STAGGER) [SQUARE]

RUN MAPS

FILES:
- IN - IMAGE.DAT
- OUT - MAPS.DAT
- IN/OUT - MSET.DAT
- USER PARAMETERS

IMAGE PARTITION:
- PIXEL INDEX DIRECTION
- LINE INDEX DIRECTION
- EACH CELL (1 2 3 4) [ALL 1]

MICRO FIDELITY CONTROL:

PARAMETERS: GROUP 1 | GROUP 2 | GROUP 3 | GROUP 4
- CONTRAST SCALE
  - TAPER (---)
  - STEP FRACTION [0.3]
  - STEP BIAS [0.1]
- OR:
  - MATRIX: E M L U E M L U E M L U E M L U
    - 0-1
    - 1-2
    - 2-3
    - 3-4
    - 4-5

GRAY-SCALE MANIPULATIONS:

- CONTRAST REMAP: CODE CONTRAST
  - PIECEWISE (0-255)
  - LINEAR (0-255)
  - BREAKPOINT PAIRS
  - (NON-DECREASING)
- CODE (0-255)
- CONTRAST (0-4095)

- INTENSITY REMAP: CODE INTENSITY
  - CODE (0-255)
  - INTENSITY (0-4095)

- INTENSITY RESET:
  - (MEAN)
  - (PSEUDO-MEDIAN)
  - (LOWEST)
  - (SECOND)
  - (THIRD)
  - (HIGHEST)

- CODE (0-255)
- INTENSITY (0-4095)
  - [MEAN]

RESULTS:
- COMPRESSION:
  - MEAN SQUARE ERROR: 2
- LEVEL:
  - 0 1 2 3 4 5 MAPSELS:
  - OPTIMAL BIASES:

II-114
<table>
<thead>
<tr>
<th>Maps Planning Form</th>
<th>(Allowed Range) [Default]</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Product Generation)</td>
<td></td>
<td>Image Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compression</td>
</tr>
</tbody>
</table>

**Module**

**User Inputs**

**Run DMaps**

**Files:**
- IN - MAPS.DAT
- OUT - DMAPS.DAT
- OUT - LEVEL.DAT

**Run Adapt** /Not with Stagger Grid/

**Files:**
- IN - DMAPS.DAT
- IN - LEVEL.DAT
- OUT - ADAPT.DAT

**Run Differ**

**Files:**
- IN - IMAGE.DAT
- IN - DMAPS.DAT
- IN - ADAPT.DAT
- OUT - ERROR.DAT
- OUT - EPRINT.DAT (Listing)

**Run Raster**

**Files:**
- IN - IMAGE.DAT
- IN - DMAPS.DAT
- IN - LEVEL.DAT
- IN - ADAPT.DAT
- IN - ERROR.DAT
- OUT - IRAST.DAT
- OUT - DRAST.DAT
- OUT - LRAST.DAT
- OUT - ARAST.DAT
- OUT - ERAST.DAT

**Maps Raster Product (Image DMaps Level Adapt Error)** [DMAPS]

**Convolution Weighting (Gaussian Uniform)** [6]

**Sigma Multiple at Window Corner [2:0]** /Gaussian Only/

**Random Dither Amplitude [2:0]**

**Difference Parameter:**
- <0 Signed =0 Statistics Only >0 Amplification Factor [10]

II-115
## Annotation and Image Assembly Planning Form

**RUN ANNOTE**

<table>
<thead>
<tr>
<th>Output Image:</th>
<th>Type</th>
<th>Printer listing</th>
<th>Lines</th>
<th>Pixels</th>
<th>Complement (Y N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Raster file)</td>
<td>RASTER</td>
<td>ANMG.DAT</td>
<td>[N]</td>
<td></td>
<td>[N]</td>
</tr>
<tr>
<td>(Print file)</td>
<td>PRINT</td>
<td>APRINT.DAT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Embedded Images:**

<table>
<thead>
<tr>
<th>Image</th>
<th>Filename</th>
<th>Input Skip Lines</th>
<th>Input Skip Pixels</th>
<th>Lines</th>
<th>Pixels</th>
<th>Output Position</th>
<th>Complement (Y N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>(N)</td>
<td>(N)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>(N)</td>
<td>(N)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Embedded Messages:**

<table>
<thead>
<tr>
<th>MSG</th>
<th>ORIENT</th>
<th>CHAR</th>
<th>SYMBOL</th>
<th>CENTER</th>
<th>TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MSG</th>
<th>ORIENT</th>
<th>CHAR</th>
<th>SYMBOL</th>
<th>CENTER</th>
<th>TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>(TBLR)</td>
<td>(1234)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DATE**

---

**Note:** The form includes fields for input/output image specifications, embedded images, and embedded messages with specific details such as symbol, center, and text.
PREFACE

This is Volume III of the Final Technical Report on Transportable MAPS Software. It constitutes the TransMAPS Program Maintenance Manual; its companion volumes contain a description of TransMAPS development (Volume I) and the TransMAPS User's Manual (Volume II). This volume is submitted in fulfillment of CDRL item A004 of Contract # F30602-80-C-0326.
TABLE OF CONTENTS

VOLUME III. TRANSMAPS MAINTENANCE MANUAL

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>TRANSPORTABLE MAPS SOFTWARE: THE SYSTEMS VIEW</td>
<td>III-1</td>
</tr>
<tr>
<td>1.1</td>
<td>Purpose</td>
<td>III-1</td>
</tr>
<tr>
<td>1.2</td>
<td>Maintenance Manual Organization</td>
<td>III-1</td>
</tr>
<tr>
<td>1.3</td>
<td>References</td>
<td>III-2</td>
</tr>
<tr>
<td>2.0</td>
<td>TRANSMAPS OVERVIEW</td>
<td>III-4</td>
</tr>
<tr>
<td>2.1</td>
<td>The TransMAPS Package</td>
<td>III-4</td>
</tr>
<tr>
<td>2.2</td>
<td>TransMAPS Structure</td>
<td>III-7</td>
</tr>
<tr>
<td>2.3</td>
<td>Host Computer Environment</td>
<td>III-7</td>
</tr>
<tr>
<td>3.0</td>
<td>TRANSMAPS FILES</td>
<td>III-10</td>
</tr>
<tr>
<td>3.1</td>
<td>User File Communication</td>
<td>III-10</td>
</tr>
<tr>
<td>3.2</td>
<td>MAPS File Formats</td>
<td>III-11</td>
</tr>
<tr>
<td>3.3</td>
<td>Standard MAPS File Header</td>
<td>III-11</td>
</tr>
<tr>
<td>4.0</td>
<td>INSTALLATION</td>
<td>III-15</td>
</tr>
<tr>
<td>4.1</td>
<td>FLX File Transport</td>
<td>III-15</td>
</tr>
<tr>
<td>4.2</td>
<td>Data Files</td>
<td>III-16</td>
</tr>
<tr>
<td>4.3</td>
<td>FORTRAN Source Code Files</td>
<td>III-17</td>
</tr>
<tr>
<td>4.4</td>
<td>FORTRAN Object Code Files</td>
<td>III-17</td>
</tr>
<tr>
<td>4.5</td>
<td>File Renaming and Protection</td>
<td>III-17</td>
</tr>
<tr>
<td>4.6</td>
<td>Program Compilation</td>
<td>III-18</td>
</tr>
<tr>
<td>4.7</td>
<td>Task Building</td>
<td>III-18</td>
</tr>
<tr>
<td>4.8</td>
<td>Test and Verification</td>
<td>III-20</td>
</tr>
</tbody>
</table>

III-i
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0</td>
<td>MODIFICATION</td>
<td>III-24</td>
</tr>
<tr>
<td>5.1</td>
<td>Default Settings</td>
<td>III-24</td>
</tr>
<tr>
<td>5.2</td>
<td>Hard-copy or Low-baud-rate Terminals</td>
<td>III-25</td>
</tr>
<tr>
<td>5.3</td>
<td>System-Specific Code</td>
<td>III-26</td>
</tr>
<tr>
<td>6.0</td>
<td>MAINTENANCE</td>
<td>III-27</td>
</tr>
<tr>
<td>6.1</td>
<td>Generic Program Structure</td>
<td>III-27</td>
</tr>
<tr>
<td>6.2</td>
<td>Integrated User Interaction</td>
<td>III-29</td>
</tr>
<tr>
<td>6.3</td>
<td>Subroutine Communication</td>
<td>III-29</td>
</tr>
<tr>
<td>6.4</td>
<td>Intra-Program Documentation</td>
<td>III-30</td>
</tr>
<tr>
<td>7.0</td>
<td>TRANSMAPS Module #1: RASTER TO SUBFRAME CONVERSION</td>
<td>III-32</td>
</tr>
<tr>
<td>7.1</td>
<td>Program Characteristics</td>
<td>III-32</td>
</tr>
<tr>
<td>7.2</td>
<td>Source Listing</td>
<td>III-32</td>
</tr>
<tr>
<td>8.0</td>
<td>TRANSMAPS MODULE #2: MAPS COMPRESSION</td>
<td>III-47</td>
</tr>
<tr>
<td>8.1</td>
<td>Program Characteristics</td>
<td>III-47</td>
</tr>
<tr>
<td>8.2</td>
<td>Source Listing</td>
<td>III-47</td>
</tr>
<tr>
<td>9.0</td>
<td>TRANSMAPS MODULE #3: MAPS DECOMPRESSION AND RESOLUTION IMAGE FORMATION</td>
<td>III-77</td>
</tr>
<tr>
<td>9.1</td>
<td>Program Characteristics</td>
<td>III-77</td>
</tr>
<tr>
<td>9.2</td>
<td>Source Listing</td>
<td>III-77</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>10.0</td>
<td>TRANSMAPS MODULE #4: MAPS ADAPTIVE IMAGE SMOOTHING......</td>
<td>III-87</td>
</tr>
<tr>
<td>10.1</td>
<td>Program Characteristics...............</td>
<td>III-87</td>
</tr>
<tr>
<td>10.2</td>
<td>Source Listing..........................</td>
<td>III-87</td>
</tr>
<tr>
<td>11.0</td>
<td>TRANSMAPS MODULE #5: MAPS DIFFERENCE IMAGE FORMATION...............</td>
<td>III-106</td>
</tr>
<tr>
<td>11.1</td>
<td>Program Characteristics...............</td>
<td>III-106</td>
</tr>
<tr>
<td>11.2</td>
<td>Source Listing..........................</td>
<td>III-106</td>
</tr>
<tr>
<td>12.0</td>
<td>TRANSMAPS MODULE #6: SUBFRAME TO RASTER CONVERSION...............</td>
<td>III-118</td>
</tr>
<tr>
<td>12.1</td>
<td>Program Characteristics...............</td>
<td>III-118</td>
</tr>
<tr>
<td>12.2</td>
<td>Source Listing..........................</td>
<td>III-118</td>
</tr>
<tr>
<td>13.0</td>
<td>TRANSMAPS MODULE #7: IMAGE ASSEMBLY AND ANNOTATION...............</td>
<td>III-128</td>
</tr>
<tr>
<td>13.1</td>
<td>Program Characteristics...............</td>
<td>III-128</td>
</tr>
<tr>
<td>13.2</td>
<td>Source Listing..........................</td>
<td>III-128</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2-1</td>
<td>TransMAPS Structure</td>
<td>III-8</td>
</tr>
<tr>
<td>4-1</td>
<td>MAPS Familiarization Test Image</td>
<td>III-22</td>
</tr>
<tr>
<td>4-2</td>
<td>MAPS Compression Logic-Diagnostic Image</td>
<td>III-23</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-I</td>
<td>Contents of TransMAPS</td>
<td>III-5</td>
</tr>
<tr>
<td>3-I</td>
<td>TransMAPS Files</td>
<td>III-12</td>
</tr>
<tr>
<td>6-I</td>
<td>Generic MAPS Program Comments</td>
<td>III-31</td>
</tr>
</tbody>
</table>
TRANSMAPS MAINTENANCE MANUAL

SECTION ONE

TRANSPORTABLE MAPS SOFTWARE: THE SYSTEMS VIEW

This document provides formal description of the Transportable MAPS Software Package or 'TransMAPS' from a systems viewpoint.

1.1 Purpose

Micro-Adaptive Picture Sequencing (MAPS) is a computationally-efficient, contrast-adaptive, variable-resolution spatial image coding technique. The TransMAPS Software Package implements the MAPS processes and related support functions in an integrated software system which is designed to be transportable among a variety of high-use mini-computers in the DEC computer family. The purpose of this implementation is to broaden access to MAPS. The ultimate intent is to establish a vehicle suitable for direct exploration of the MAPS technique and to provide a system capable of supporting functional application of MAPS to real image coding tasks.

The current document addresses those areas specifically concerned with the relation of TransMAPS to its host computer system. These include software installation, modification, and maintenance activities. The intended audience consists of systems personnel charged with support of TransMAPS at their site and others who wish to understand MAPS at a detailed level of 'how it's done'.

1.2 Maintenance Manual Organization

The information necessary for initial installation of TransMAPS at a new
site is given in Sections Two, Three, and Four. Section Two provides an overview of TransMAPS and the conditions assumed for the host computer environment. Section Three describes the extensive file system which supports TransMAPS and specifies the requirements imposed on communication of user image files. Section Four deals specifically with questions of TransMAPS installation from its source on computer-compatible magnetic tape.

Sections Five and Six discuss more details on the internal structure of the TransMAPS process modules. Section Five addresses questions of program modification which may arise in refining TransMAPS for a particular site. Section Six outlines general program structure conventions which should provide helpful guides in subsequent program maintenance activities.

Finally, Sections Seven through Thirteen contain complete COMMENT-annotated program listings of the TransMAPS source code. These listings have had running titles appended which give both main module and current-routine names. This should be an aid to rapid location of the relevant section for this deepest level of detailed documentation.

1.3 References

The material on initial installation in this document is intended as a self-contained path to achieving a functioning TransMAPS system. Beyond this however, at the level of program modification or maintenance, it is assumed that the reader is familiar with both the conceptual basis of MAPS and the structure of the user option space in the TransMAPS package. This material is described in the TransMAPS User's Manual to a level adequate for most purposes. For more detailed information on specific MAPS concepts and processes, the reader is referred to the following collection of documents:
References:

RADC MAPS-Related Reports:


SPIE Proceedings Articles:


III-3
SECTION TWO

TRANSMAPS OVERVIEW

This section outlines the contents of the TransMAPS package, its structure, and the assumed host computer environment.

2.1 The TransMAPS Package

The contents of the TransMAPS package are listed in Table 2-I.

The seven main program modules are provided on computer-compatible magnetic tape in the form of both source code and object code files. The first six modules relate directly to MAPS processes. Module #7 provides a stand-alone image assembly and annotation capability. This seventh module supports display of MAPS product images with assembly of comparative image frames and corresponding identifying annotation. It can, however, also be used with other user imagery to integrate and label related frames for various forms of presentation.

The six MAPS modules further subdivide into two functional classes. The first two relate to the MAPS compression phase. Module #1 converts the raster source imagery to the appropriate subframe organization for MAPS processing. Module #2 implements the actual MAPS micro-coding technique on a subframe by subframe basis. This second module contains the most complex portions of the interactive user option space.

The next four modules deal with MAPS product formation from the compressed image stream. Module #3 reconstructs the tonal image - MAPS decompression - and simultaneously generates a MAPS 'resolution image' which displays the micro-adaptive variable resolution created by MAPS.
TABLE 2-I. CONTENTS OF TransMAPS

TransMAPS Package:

Seven Main Program Modules:

<table>
<thead>
<tr>
<th>Module</th>
<th>Functional Description</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 SUBFRM</td>
<td>Raster to Subframe Conversion</td>
<td>SF.FSV, SF.OBJ</td>
</tr>
<tr>
<td>02 MAPS</td>
<td>MAPS Compression</td>
<td>MP.FSV, MP.OBJ</td>
</tr>
<tr>
<td>03 DMAPS</td>
<td>MAPS Decompression &amp; Level Image Formation</td>
<td>DN.FSV, DN.OBJ</td>
</tr>
<tr>
<td>04 ADAPT</td>
<td>MAPS Adaptive Image Smoothing</td>
<td>AD.FSV, AD.OBJ</td>
</tr>
<tr>
<td>05 DIFFER</td>
<td>MAPS Difference Image Formation</td>
<td>DF.FSV, DF.OBJ</td>
</tr>
<tr>
<td>06 RASTER</td>
<td>Subframe to Raster Conversion</td>
<td>RS.FSV, RS.OBJ</td>
</tr>
<tr>
<td>07 ANNOTE</td>
<td>Image Assembly and Annotation</td>
<td>AI.FSV, AI.OBJ</td>
</tr>
</tbody>
</table>

Provided on Computer-Compatible Tape (CCT):

FORTRAN IV-Plus Source Code - x.FSV
FORTRAN Object (F4P) Code - x.OBJ

Six Data Sets:

<table>
<thead>
<tr>
<th>Description</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation Symbol-Map Tables</td>
<td>SYMBOL.BIN</td>
</tr>
<tr>
<td>MAPS Compression Test Image (160 x 128)</td>
<td>MTEST.BIN</td>
</tr>
<tr>
<td>Sample MAPS User Parameter Set (Use with MTEST)</td>
<td>MSET.BIN</td>
</tr>
<tr>
<td>MAPS Product Generation Test Image (120 x 128)</td>
<td>GIRL6.BIN</td>
</tr>
<tr>
<td>Two &quot;video Frame&quot; Images (480 lines x 624 pixels):</td>
<td></td>
</tr>
<tr>
<td>Building Scene</td>
<td>BLDGIMG.BIN</td>
</tr>
<tr>
<td>IEEE Girl</td>
<td>GIRLIMG.BIN</td>
</tr>
</tbody>
</table>

Provided on Computer-Compatible Tape (CCT)

MAPS File Structure with Standard Filenames and Headers:

<table>
<thead>
<tr>
<th>Description</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source image (One Subframe/FIXED Record)</td>
<td>IMAGE.DAT</td>
</tr>
<tr>
<td>User Parameter Set</td>
<td>MSET.DAT</td>
</tr>
<tr>
<td>MAPS Compression Stream (FIXED Records)</td>
<td>MAPS.DAT</td>
</tr>
<tr>
<td>MAPS Block/Pattern Image (Subframes)</td>
<td>DMAPS.DAT</td>
</tr>
<tr>
<td>MAPS Resolution Image (Subframes)</td>
<td>LEVEL.DAT</td>
</tr>
<tr>
<td>MAPS Adaptively Smoothed Image (Subframes)</td>
<td>ADAPT.DAT</td>
</tr>
<tr>
<td>MAPS Difference Image (Subframes)</td>
<td>ERROR.DAT</td>
</tr>
<tr>
<td>Fidelity Performance Summary (Listing)</td>
<td>EPRINT.DAT</td>
</tr>
<tr>
<td>MAPS Product Image (Raster)</td>
<td>XRAST.DAT</td>
</tr>
<tr>
<td>MAPS Product Image (Listing)</td>
<td>XRIST.DAT</td>
</tr>
<tr>
<td>Annotation Symbol-Map Tables</td>
<td>SYMBOL.DAT</td>
</tr>
<tr>
<td>Annotated Image (Raster)</td>
<td>ANIMG.DAT</td>
</tr>
<tr>
<td>Annotated Printer Pseudo-Image (Listing)</td>
<td>APRIN.T.DAT</td>
</tr>
</tbody>
</table>

User Interaction Pre-Planning Aids:

<table>
<thead>
<tr>
<th>Description</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPS Planning Form - Compression</td>
<td></td>
</tr>
<tr>
<td>MAPS Planning Form - Product Generation</td>
<td></td>
</tr>
<tr>
<td>Annotation and Image Assembly Planning Form</td>
<td></td>
</tr>
</tbody>
</table>

Documentation:

<table>
<thead>
<tr>
<th>Description</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransMAPS User's Manual</td>
<td></td>
</tr>
<tr>
<td>TransMAPS Maintenance Manual</td>
<td></td>
</tr>
</tbody>
</table>
Module #4 further refines the decompressed image through a MAPS-based adaptive smoothing process. Module #5 evaluates the fidelity of the MAPS coding through formation of a difference image between the source and product images. Finally, Module #6 converts any of the subframe organized MAPS images back to raster format for interface with the outside world.

The seven program modules are supported by six data files. The first, file SYMBOL, contains the bit maps which make up the source data for the annotation symbol set. Each of the sixty symbols is represented by a 48x48 bit map packed as forty-eight lines of three sixteen-bit words each. Bit packing is left to right within each word. Once established on the system, this file is read in automatically when Module #7, ANNOTE, is invoked. The annotation symbols are then generated as needed by resampling these bit maps to the user-requested size.

Two 'toy' size images - files MTEST and GIRL6 - are supplied with the package. These images are designed for rapid testing of the modules and to aid with inexpensive exploration of the space of user options. A sample set of pre-defined user parameters - file MSET - is also provided; it is set up to be used with the MTEST image.

Finally, two 'real world' (video frame size) images - GIRLIMG and BLDGIMG - are included to provide more realistic test examples.

Intermodule communication is provided by a set of twelve standard MAPS files and is designed to be transparent to the user. Dedicated mnemonic file names are used, however, so that both user and systems personnel can easily assess status of the process following interruption or deliberate suspension of a MAPS interactive task.
The user is also aided by a series of three 'planning forms' which display the space of user options schematically. These forms have spaces for entry of user-selected options to assist in pre-planning and documenting a MAPS interactive session. Moreover, the planning forms provide a convenient guide to the location of various user option interactions for the use of systems personnel.

The TransMAPS package is completed by two volumes of formal documentation - The TransMAPS User's Manual and this Maintenance Manual.

2.2 TransMAPS Structure

The previous section provides only a tabular listing of the contents of the TransMAPS package. The key structure of TransMAPS is presented in Figure 2-1. This diagram shows the interrelationships among all of the process modules and data files. A brief descriptor of the contents of each file is given along with the corresponding standard file name. Figure 2-1 is the primary portrayal of the overall organization of TransMAPS.

2.3 Host Computer Environment

TransMAPS is intended to run on DEC PDP-11/45, PDP-11/70, and VAX computer systems. It was targeted specifically at the PDP-11/70 in the Image Processing System at RADC. The modules are all written in DEC's FORTRAN IV-RLUS. They are each designed to run under DEC's RSX-11M Operating System as single task loads without overlays. The file systems are such that they should be compatible with either RMS-11 or FCS-11.

The TransMAPS modules do not make use of the VIRTUAL declaration construct so a 'mapped' memory system should not be necessary. Moreover, the restriction to FORTRAN IV-PLUS is thought to involve only the use of
Figure 2-1. TransMAPS Structure
the IISHFT library shift function in Module #7 and the use of Integer*4 arithmetic for a few performance statistics and element count accumulations. At some penalty in execution efficiency, it should thus be possible to modify the code to be compatible with the more restricted DEC FORTRAN compiler if necessary at a particular site.

The user interaction portion of the code tacitly assumes a high-speed channel to the interactive terminal. Thus, some of the query prompts update rather extended text with each user response to provide complete and immediate feedback. If much slower terminal channels (e.g. low-speed telephone modem) or hardcopy terminals are to be employed, these sections of the interaction can be slightly modified to reduce the text redundancy. Guidelines for such changes are provided in Section Five.
SECTION THREE

TRANSMAPS FILES

This section discusses user image file conventions, the formats of standard MAPS files, and the contents of the MAPS file header.

3.1 User File Communication

DEC FORTRAN does not support direct magnetic tape access in any convenient way. Thus, user-supplied source imagery must be entered as a disk file for communication to TransMAPS. Furthermore, there does not appear to be any standard format for image data files which extends across the systems on which TransMAPS is to be used. Thus, the file format for TransMAPS input was chosen to minimize the buffer overhead incurred in the program modules.

TransMAPS assumes that user raster source images will be supplied as sequential binary files written using the SEGMENTED record type. This form allows the physical record size (and consequently the buffer) to be much smaller than the largest logical record size expected. Moreover, the physical records are sufficiently small that file transfer utilities such as FLX can be employed. Because the types of source formatting for a particular site or application are unknown, responsibility for provision of the conversion utilities to establish the SEGMENTED source raster files is presumed to reside with each site. For test purposes, the imagery delivered as part of the TransMAPS package has already been placed in this form. Thus initial installation and checkout can proceed independently from the generation of this image conversion capability. TransMAPS does accommodate both eight bit per pixel and six bit per pixel source forms. For the six bit sources, the so-called DICOMED format convention is assumed; namely, six-bit pixels right justified in eight-bit fields. This convention avoids the 'byte-swap' complications.
encountered between DEC (lowest byte first) and many other computer systems (most significant byte first). This six bit convention is also the standard for such imagery at RADC.

3.2 MAPS File Formats

File formats for the remaining MAPS files are determined by access type and file content. Printer listing files are, of course, of FORMATTED type. All raster image files (input or output) are of SEGMENTED binary type. Subframe-organized image files are of FIXED record length binary type suitable for DIRECT access under either FCS-11 or RMS-11 file systems. The MAPS compressed image stream is also written as a FIXED record length binary file with records of 512 bytes. Finally, files containing user parameter sets and the symbol table bit maps are expressed as SEGMENTED binary files.

All MAPS data files with their mnemonic file names, content descriptors, and file type characteristics are listed in Table 3-1.

3.3 Standard MAPS File Header

The various 'internal' MAPS image files - IMAGE, MAPS, DMAPS, LEVEL, ADAPT, and ERROR - all require certain ancillary information in their roles of intermodule communication. This information is carried in a standard MAPS file header which is the first record in each file. In each case, the record length is adjusted to match the corresponding FIXED length for the file. Thus, the header content must fit within the smallest allowed value of this record length - namely, 64 bytes.
### Table 3-I. TransMaps Files.

**File Characteristics:**

<table>
<thead>
<tr>
<th>Filename</th>
<th>Content</th>
<th>Filetype</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Source Image, Raster</td>
<td>SEGMENTED</td>
</tr>
<tr>
<td>IMAGE</td>
<td>Source Image, Subframes</td>
<td>FIXED *</td>
</tr>
<tr>
<td>MSET</td>
<td>Maps User Parameter Set</td>
<td>SEGMENTED</td>
</tr>
<tr>
<td>MAPS</td>
<td>Maps Compression Stream</td>
<td>FIXED **</td>
</tr>
<tr>
<td>DMAPS</td>
<td>Maps Block/Pattern, Subframes</td>
<td>FIXED *</td>
</tr>
<tr>
<td>LEVEL</td>
<td>Maps Resolution Image, Subframes</td>
<td>FIXED *</td>
</tr>
<tr>
<td>ADAPT</td>
<td>Maps Adaptively Smoothed, Subframes</td>
<td>FIXED *</td>
</tr>
<tr>
<td>ERROR</td>
<td>Maps Difference Image, Subframes</td>
<td>FIXED *</td>
</tr>
<tr>
<td>EPRINT</td>
<td>Fidelity Performance Listing</td>
<td>FORMATTED</td>
</tr>
<tr>
<td>IRAST</td>
<td>Source Image, Raster</td>
<td>SEGMENTED</td>
</tr>
<tr>
<td>DRAST</td>
<td>Maps Decompressed Image, Raster</td>
<td>SEGMENTED</td>
</tr>
<tr>
<td>LRAST</td>
<td>Maps Resolution Image, Raster</td>
<td>SEGMENTED</td>
</tr>
<tr>
<td>ARAST</td>
<td>Maps Adaptively Smoothed, Raster</td>
<td>SEGMENTED</td>
</tr>
<tr>
<td>ERAST</td>
<td>Maps Difference Image, Raster</td>
<td>SEGMENTED</td>
</tr>
<tr>
<td>SYMBOL</td>
<td>Annotation Symbol-Map Tables</td>
<td>SEGMENTED</td>
</tr>
<tr>
<td>User</td>
<td>Embedded Input Image, Raster</td>
<td>SEGMENTED</td>
</tr>
<tr>
<td>ANIMG</td>
<td>Annotated Image, Raster</td>
<td>SEGMENTED</td>
</tr>
<tr>
<td>APRINT</td>
<td>Annotated Pseudo-Image Listing</td>
<td>FORMATTED</td>
</tr>
</tbody>
</table>

* "FIXED" Subframe records are 64, 256, or 1024 bytes

**Fixed** Compression Stream Records are 512 bytes
Currently, the header information occupies the first fifty of these locations and is organized as follows:

**Standard MAPS File Header:**

<table>
<thead>
<tr>
<th>Identification:</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File Type</strong></td>
<td></td>
</tr>
<tr>
<td>0: IMAGE</td>
<td>2</td>
</tr>
<tr>
<td>1: MAPS</td>
<td></td>
</tr>
<tr>
<td>2: DMAPS</td>
<td></td>
</tr>
<tr>
<td>3: LEVEL</td>
<td></td>
</tr>
<tr>
<td>4: ADAPT</td>
<td></td>
</tr>
<tr>
<td>5: ERROR I-D</td>
<td></td>
</tr>
<tr>
<td>6: ERROR I-A</td>
<td></td>
</tr>
<tr>
<td>7: ERROR D-A</td>
<td></td>
</tr>
<tr>
<td><strong>Image Name</strong></td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image Size:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines/Image</td>
<td>2</td>
</tr>
<tr>
<td>Pixels/Line</td>
<td>2</td>
</tr>
<tr>
<td>Bits/Pixel</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image Partition:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Subframe Size (0 16 32)</td>
<td>2</td>
</tr>
<tr>
<td>Subframe Grid (0=Square, 1=Stagger)</td>
<td>2</td>
</tr>
<tr>
<td><strong>Subframe Count</strong></td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAPS Results:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MAPSel Count</strong></td>
<td>4</td>
</tr>
<tr>
<td>Block/Pattern Vector (1 Bit/Level, Packed Right to Left)</td>
<td>2</td>
</tr>
<tr>
<td><strong>Optimal Pattern Biases</strong> (Level [&gt;0], Low/High)</td>
<td>5x2x2 = 20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Future Use:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>14</strong></td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL 64**

Thus, there are fourteen additional bytes available for future expansion before a more elaborate header structure must be sought. Note that a separate record is used for the header on the MAPS-compressed data stream. This, however, is not really necessary since the division into 512-byte records is arbitrary and asynchronous. The MAPS file header could be incorporated into the first fifty bytes of the first record followed immediately by the start of the MAPSel data stream. Only slight modification to the initialization procedures would be required to accomplish this. It can be done in situations where minimum compressed data size is sought.

III-13
As might be expected, almost all information in the header is created in the first two modules - SUBFRM and MAPS. The first two bytes encode the type of data in the file. This preserves the integrity of the file contents through subsequent file name changes. The next twenty-two bytes describe the source image and its partition into subframes in program SUBFRM. The remaining twenty-six bytes characterize the MAPS compression and the optimum pattern bias values needed for decompression. This data is generated in Module #2, MAPS. The unused header bytes might be employed in the future to carry the control selections used to create alternate product images.
SECTION FOUR

INSTALLATION

This section describes the procedures required to transfer TransMAPS from its source computer-compatible magnetic tape to a functional system.

4.1 FLX File Transport

The TransMAPS software and support data resides on nine-track magnetic tape (density 800 bpi) in the form of twenty FLX files. These are further subdivided into three file categories - seven FORTRAN source code files, seven FORTRAN object code files, and six data files.

The command sequence to enter these files on the system is as follows:

```
ALL MMu: Allocate tape drive
FLX MMu:[50,27]/LI List FLX tape contents (optional)
FLX SY:[g,m]/RS = MMu:[50,27]*.*/DO Transfer all TransMAPS files
PIP/LI List transferred files (optional)
DEA MMu: Deallocate tape drive
```

Here, 'u' is the physical unit number of the tape drive on which the TransMAPS FLX tape is mounted and [g,m] is the UIC (user identification code) under which the TransMAPS software is to be installed. The UIC under which the system was originally transferred to tape was [50,27].
The file names and sizes which should appear in the directory listings are as follows:

<table>
<thead>
<tr>
<th>File</th>
<th>Content</th>
<th>FLX size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF.FSV</td>
<td>SUBFRM FORTRAN Source Code</td>
<td>29</td>
</tr>
<tr>
<td>MP.FSV</td>
<td>MANS FORTRAN Source Code</td>
<td>67</td>
</tr>
<tr>
<td>DM.FSV</td>
<td>DMAPS FORTRAN Source Code</td>
<td>20</td>
</tr>
<tr>
<td>AD.FSV</td>
<td>ADAPT FORTRAN Source Code</td>
<td>43</td>
</tr>
<tr>
<td>DF.FSV</td>
<td>DIFFER FORTRAN Source Code</td>
<td>28</td>
</tr>
<tr>
<td>RS.FSV</td>
<td>RASTER FORTRAN Source Code</td>
<td>23</td>
</tr>
<tr>
<td>AI.FSV</td>
<td>ANNOTE FORTRAN Source Code</td>
<td>61</td>
</tr>
<tr>
<td>SF.OBJ</td>
<td>SUBFRM Object Code</td>
<td>26</td>
</tr>
<tr>
<td>MP.OBJ</td>
<td>MANS Object Code</td>
<td>71</td>
</tr>
<tr>
<td>DM.OBJ</td>
<td>DMAPS Object Code</td>
<td>30</td>
</tr>
<tr>
<td>AD.OBJ</td>
<td>ADAPT Object Code</td>
<td>38</td>
</tr>
<tr>
<td>DF.OBJ</td>
<td>DIFFER Object Code</td>
<td>25</td>
</tr>
<tr>
<td>RS.OBJ</td>
<td>RASTER Object Code</td>
<td>17</td>
</tr>
<tr>
<td>AI.OBJ</td>
<td>ANNOTE Object Code</td>
<td>67</td>
</tr>
<tr>
<td>SYMBOL.BIN</td>
<td>Symbol Map Tables</td>
<td>37</td>
</tr>
<tr>
<td>GIRL6.BIN</td>
<td>Girl 6-bit 'Toy' Image</td>
<td>34</td>
</tr>
<tr>
<td>MTEST.BIN</td>
<td>MAPS Logic Test 'Toy' Image</td>
<td>46</td>
</tr>
<tr>
<td>MSET.BIN</td>
<td>Sample User Parameter Set</td>
<td>2</td>
</tr>
<tr>
<td>BLDGIMG.BIN</td>
<td>'Building Scene' Video Image</td>
<td>631</td>
</tr>
<tr>
<td>GIRLIMG.BIN</td>
<td>'IEEE Girl' Video Image</td>
<td>631</td>
</tr>
</tbody>
</table>

4.2 Data Files

The six TransMAPS data files are all transferred under file type `.BIN`. Each of these files is in SEGMENTED binary format. The choice of the `.BIN` type allows FLX to default automatically to Formatted Binary mode. Note that in the FLX tape versions of the files, binary headers and checksums are added to the data. This means that the FLX file size for such files will typically be somewhat larger than the corresponding Files-11 versions. These differences will be reflected in the file sizes reported when PIP is used to list the directory after file transfer. In the Files-11 disk environment, for example, the 'video' image sizes are each 608 blocks.
4.3 FORTRAN Source Code Files

The FORTRAN source code for the seven TransMAPS modules was transferred with an arbitrary file type of '.FSV'. This was done to permit retention of the original version of the source code if later versions are created and then all but the most recent eliminated by a PIP purge switch.

4.4 FORTRAN Object Code Files

The object code for the seven TransMAPS modules was produced by the FORTRAN IV-PLUS compiler (F4P) under RSX-11M on a PDP-11/70. Here, the standard file type, '.OBJ' was used for the transfer. This file type is another of the small class of names for which FLX correctly defaults to the Formatted Binary mode.

4.5 File Renaming and Protection

Note that for both the source and object code files, the file names have been shortened to two-character mnemonics. Experienced users will find these short names useful in reducing the keystrokes needed to invoke the processes during interactive execution. Inexperienced users, however, will probably prefer the security of the longer mnemonics. Thus, it is suggested that the '/EN' switch be used with PIP to enter both the full and short versions of the file name as synonyms in the file directory. This probably need be done only for the task files, type '.TSK', after they have been generated.

The program modules will expect the data files to exhibit type '.DAT'. Thus, the '.BIN' files should all be renamed using the PIP switch '/RE'. Alternatively, the '.BIN' files can be retained as backup and copies created by PIP with a file type of '.DAT'.
In addition to or as an alternative strategy to protecting the key TransMAPS files by using multiple copies with different names, the explicit protection status of the files can be changed via the PIP switch '/FR' with appropriate subswitches. The source code files and the critical data files (SYMBOL.DAT, MTEST.DAT, and GIRL6.DAT) can all be changed to read-only through the switch/subswitch sequence '/FR/SY:R/OW:R/GR:R/WO:R'. The sample user parameter set file, MSET.DAT, can be given deletion protection by replacing the ':R' entries with ':RWE'. Read-only protection can also be applied to the '.TSK' task files once they have been built into the system. Such protection redefinition should help to insure that the functional package is not inadvertently deleted during file maintenance operations where 'wild cards' are sometimes used in some of the file specification fields.

4.6 Program Compilation

In principle, task building could proceed directly from the set of TransMAPS object code files provided as part of the system. At new sites, however, it is probably safer to recompile the FORTRAN source code. This is done for each of the seven modules by the command line:

    F4P File.OBJ,File.LST=File.FSV or F4P File,File=File.FSV.

Here, 'File' takes on the 'values' SF, MP, DM, AD, DF, RS, and AI. Note that this compilation string produces a source code listing with a cross-reference map as well as the object code file for each module.

4.7 Task Building

Following compilation, each module must be 'task built' into a loadable form. Task building of most of the TransMAPS modules also requires exercise of some of the 'TKB' options to change various default settings.
The general sequence and the specific options are outlined in the following tabulation:

**Task Build Options:**

**General Sequence:**

```
>TKB (cr)
TKB> task/FP,map=object (cr)
TKB> / (cr)
ENTER OPTIONS:
TKB> option1=n1 (cr)
TKB> option2=n2 (cr)
TKB> // (cr)
```

<table>
<thead>
<tr>
<th>Program Module</th>
<th>Options Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBFRM (SF)</td>
<td>MAXBUF=1024</td>
</tr>
<tr>
<td>MAPS (MP)</td>
<td>MAXBUF=1024</td>
</tr>
<tr>
<td>DMAPS (DM)</td>
<td>MAXBUF=1024</td>
</tr>
<tr>
<td>ADAPT (AD)</td>
<td>MAXBUF=1024</td>
</tr>
<tr>
<td>DIFFER (DF)</td>
<td>MAXBUF=1024, ACTFIL=5</td>
</tr>
<tr>
<td>RASTER (RS)</td>
<td>MAXBUF=1024</td>
</tr>
<tr>
<td>ANNOTE (AI)</td>
<td>None</td>
</tr>
</tbody>
</table>

Note that all but one of the options involves increasing the maximum file buffer size to accommodate the largest expected FIXED record length in the subframe-organized files. The ACTFIL increase in the DIFFER module (from the default value of four) is required by the addition of the performance evaluation listing, EPRINT.DAT, to the normal set of image data files.

It might also be noted that a shorter task build command can be used with the ANNOTE module since no TKB options are required there. The appropriate command line takes the following form:

```
TKB AI.TSK/FP,AI.MAP=AI.OBJ or TKB AI/FP,AI=AI.
```
The '/FP' switch which shows up on the task file in all cases here is used to invoke the 'Floating Point Processor'. Under some versions of RSX-11M, this switch is required even if the module contains no instructions involving floating point processes. Failure to include this switch is not detected at task build but generates run time error #2: "TASK INITIALIZATION FAILURE".

4.8 Test and Verification

Once the seven modules have undergone task building and the data files have been renamed (or copied) to type '.DAT', the TransMAPS software should be ready for application. Several quick tests can be run to verify the system. A suggested sequence to check the interface modules proceeds as follows:

- RUN SF on GIRL6.DAT to convert the 120 line by 128 pixel by 6 bit raster image to 8 bit subframe form;
- RUN RS on IMAGE.DAT to convert the resultant subframe image back to raster organization; and
- RUN AI on IRAST.DAT with annotation and in the pseudo-image mode to replicate the output shown in Figure 4-1.

This process will verify operation of these three modules and will exercise the SYMBOL.DAT bit map tables. It also provides immediate visual confirmation via the line printer even on systems where no other image display is available. Note that the printer should be switched to an eight-line-per-inch mode to more closely approximate square pixels (an 8:10 ratio rather than 6:10).

The IMAGE.DAT file for the girl 'toy' image can also be used to test the various processes directly related to MAPS. Here it is suggested that
some or all of the examples in Section Nine of the User's Manual be reproduced. Because of the small image size, such runs go very quickly and provide immediate feedback via the printer.

Finally, the detailed MAPS logic can be verified with the diagnostic image MTEST.DAT as displayed in Figure 4-2. Examples of the use of this image are presented in Section Eight of the User's Manual. The sample user parameter set on file MSET.DAT can be used in conjunction with this image to check and demonstrate the macro-fidelity control capabilities of TransMAPS. The user options can be reviewed and updated to implement various combinations of the following diagnostic tests:

**Compression-Logic Diagnostic Tests:**

**Image Partition and Macro-Fidelity Control:**
- Input Image Line and Pixel Skips
- Subframe Phasing (Square and Staggered)
- Macro-Partition Group Assignment

**Micro-Fidelity Control:**
- Zigzag Sequencing
- Contrast Control as a Function of Transition Level
- Contrast Control as a Function of Contrast Type
- Pattern Code Assignment

**Gray-Scale Manipulation:**
- Contrast Space Quad Sort
- Intensity Space Quad Sort
- Intensity Reset Assignment

Successful completion of such checks should then insure an operational MAPS compression capability suitable for further exploration or functional image data base preparation.
Figure 4-1. MAPS Familiarization Test Image
Figure 4-2. MAPS Compression-Logic Diagnostic Image

III-23
SECTION FIVE

MODIFICATION

This section discusses three known circumstances where TransMAPS modifications might be desired - default parameter changes, accommodation of interactive terminal limitations, and system-specific code constraints.

5.1 Default Settings

TransMAPS contains an extensive space of user options and each option generally has some preset default selection. As experience with the package grows or as typical image characteristics emerge at a particular site, it may prove advantageous to change some of these default parameter settings. For example, it may be found that a particular image frame size is encountered on a regular basis. In this instance, the default line count and pixel count for the source imagery might be set to this size. As another example, a different standard subframe partition may be best for a large class of images and the corresponding controls can be selected for the default. Or again, a different set of contrast control generator parameters (taper, step fraction, or step bias) may be desired for 'normal' operation.

In order to support such changes, the default parameters for each module have been collected and placed together near the beginning of the user interaction subroutine for that program. Moreover, to make these default settings easy to locate, the relevant program lines have been 'flagged' with on-line COMMENTS of the form '! Default'. In order to effect a default change, the system personnel need merely isolate the appropriate setting in the source code, modify its value, recompile the module, and rebuild the resultant task.
5.2 Hard-copy or Low-baud-rate Terminals

TransMAPS has been implemented to give extensive feedback in the user interaction by providing immediate update and display of user option selections. In addition, extended prompts have been used to remind the user of the range of choices available at various points. This strategy is very helpful in a system with fast communication channels and video rate terminal displays. However, slow channels such as low-speed phone lines or hard-copy terminals such as tele-type or DEC-writer are less satisfactory for this approach because of their long response time.

To reduce user frustration in such environments, it is probably desirable to restructure the interaction to provide updates or complete prompts less often. Reduced frequency of output is recommended in the following three areas:

In MAPS - Present the updated image macro-fidelity partition matrix only after the user has signalled that all desired line-editing is complete;

In MAPS - Present the updated micro-fidelity control parameters and corresponding threshold matrix only after a user signal that editing is complete; and

In ANNOTE - Present the allowed character set only once before the definition of the first annotation message, not as a prompt to every message text definition.

Other user communications which could be shortened are those in ANNOTE which define the message orientation and the message length. Note that in the slowest interaction - the hard-copy terminal - the prior copy itself is available for review. Thus, reissue of prompts is required much less than in a faster but more volatile soft-copy environment.
5.3 System-Specific Code

Two known constructs specific to FORTRAN IV-PLUS were used in TransMAPS. If it is necessary to transport the package to a system where only the FORTRAN compiler is supported, these constructs would have to be simulated by less efficient code.

The first limitation is restricted to Module #7, ANNOTE, where the library shift function, IISHFT, is used in the symbol bit-map resampling processes. The shifts could be replaced by multiplies and divides with powers of two. However, the bit-map tables should probably be redefined as forty-eight lines of six bytes each, rather than as forty-eight lines of three sixteen-bit words each. This is the first step in circumventing the problems of sign interpretation and overflow in the high-order bits of the Integer*2 words. Note that each byte should first be transferred to the lower portion of a normal integer (two bytes) and sign-corrected before any of the arithmetic operations (pseudo-shifts) are carried out. Note also that the 'byte-swap' problem must be accommodated since the bit-maps are stored left to right in the original two-byte words.

The second construct involves use of Integer*4 arithmetic at several points where performance statistics or element counts are accumulated. Since the ordinary FORTRAN compiler supports the Integer*4 data type but only allows Integer*2 arithmetic on such variables, all Integer*4 arithmetic constructs must be converted to compound Integer*2 processes with extensive overflow checking. Should such conversions be needed, an attempt has been made to flag all lines containing Integer*4 arithmetic with on-line COMMENTs of the form ':: I*4'. These flags should at least help to localize the conversion process.
SECTION SIX

MAINTENANCE

This section describes common features among the program modules in order to provide a framework for understanding the overall philosophy of implementation used in TransMAPS. This information should provide useful guidance for any necessary software maintenance activities on the package.

6.1 Generic Program Structure

All seven principal TransMAPS modules have essentially the following overall structure:

Generic Program Structure:

User Interaction - Subroutine USERx (x = I,M,D,A,E,R)
File Establishment - Subroutine FILESx (x = I,M,D,A,E,R)
Process Initialization - Subroutine SETUPx (x = I,M,D,A,E,R)
PRINCIPAL LOOP on Subframes or Lines

Each program contains three main preprocess steps - user option interaction, opening and positioning of files, and initialization of tables and variables. The names for the primary subroutines which implement these preprocesses also exhibit a consistent convention - USERx, FILESx, and SETUPx. Here, 'x' is a single-character mnemonic for the host module which calls the routine. Note that the calling order for these preprocesses does vary among the modules since the file header information is needed to direct the user interaction in some cases, while the user interaction selects the relevant files in others. Such usage is clear from context in the actual program listings.
The body of each program is then typically a loop which iterates on lines or subframes within the image(s). Both the preprocesses and the body of each program are further modularized at points of natural process division. This partitions the implementation into 'graspable' chunks with an average routine length of less than fifty lines including continuations but excluding COMMENTS. Both the number of subroutines and the overall task length for each module are listed in the following tabulation:

<table>
<thead>
<tr>
<th>Module Characteristics:</th>
</tr>
</thead>
<tbody>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>Program</td>
</tr>
<tr>
<td>SUBFRM</td>
</tr>
<tr>
<td>MAPS</td>
</tr>
<tr>
<td>DMAPS</td>
</tr>
<tr>
<td>ADAPT</td>
</tr>
<tr>
<td>DIFFER</td>
</tr>
<tr>
<td>KASTER</td>
</tr>
<tr>
<td>ANNOTE</td>
</tr>
</tbody>
</table>

Note that with the exception of Module #3, DMAPS, all of the programs use a large fraction of the available 32K task space. Indeed, three of the modules just fit within the constraint imposed by the sixteen bit word length.
6.2 Integrated User Interaction

In general, the user interaction portions of the code were not further partitioned into smaller chunks. In this area, it was felt that the flow of the interaction was easier to follow if it were not distributed among several small routines.

The sequence of interactions tends to exhibit a natural punctuation. Typically, the definition of each user option or parameter proceeds through a sequence of fairly well-defined steps. First, the current value (initially the default) is checked for consistency with parameters defined previously in the interaction and reset as necessary. Next, the user is prompted with the parameter under consideration, its current value, and its current allowed range (if applicable). This prompt is in the form of a query to which the user can respond with either a new value or 'no change' as desired. The user's response is then used to update the parameter and it is again checked and corrected for consistency with the allowed range. The interaction then proceeds to the next option or parameter. Thus, the code comes naturally in a sequence of easily grasped packets.

In the TransMAPS implementation of the user interactions, the packets are further set out by indenting the consistency checking operations. Moreover, the FORMAT statements containing the query communications are placed at the location of the corresponding packets (rather than being collected at the beginning or end of the routine). These FORMAT statements then serve as integral documentation of the interaction and the flow of code need not be interrupted with separate COMMENTs.

6.3 Subroutine Communication

Modularization in MAPS is intended more to give conceptual organization
to the process than to isolate multiple-use segments of code. The flow is characterized by a sequence of processes on a common body of data using common control parameters. Thus, formal parameters to accommodate varying points of application are not needed for subroutine transfer. The data and control parameters are then conveniently organized into named COMMON blocks for interroutine communication.

These labeled COMMON blocks serve to give further structure to the process and to provide mnemonic groupings in the data and parameter spaces. Indeed, such blocking provides the organizing principle for construction of an effective DATA DICTIONARY for each module. This information is then documented by including it in the program COMMENTS.

6.4 Intra-Program Documentation

For a capable programmer, the flow of code itself is the key documentation in a program and extensive in-line COMMENTS often prove distracting. Hence, a consistent pattern of header COMMENT blocks has been adopted for the internal documentation of the TransMAPS modules. An extended block is given at the beginning of each of the seven main modules with abbreviated headers given in each of the other routines. The generic formats for these headers are summarized in Table 6-I.

Note that the main block contains a brief process descriptor, file communication definitions, a user interaction outline, a 'structured' process hierarchy, and a data dictionary. The subroutine headers contain a brief description of purpose plus the CALLing links to and from the routine.

The remaining seven sections of this manual present the detailed listings of this annotated TransMAPS source code.
Table 6-I. GENERIC MAPS PROGRAM COMMENTS

COMMENT Formats:

Principal Module Headers:

| TransMAPS Module n: process descriptor |

Control Data Corporation - 1982

Files: Unit Name Content From/To Type

In/Out n zzzzz descriptor module SEGMENTED or DIRECT or FIXED SEQ. or FORMATTED

User Interaction:

principal interactive parameter groups

Program Structure:

subroutine calling hierarchy with brief process outline

COMMON Block Communication:

/blkname/ descriptor length (1=2 words)

host routine names

variable (datatype) descriptor

(These lists provide a module DATA DICTIONARY)

order conventions or geometry definitions (if appropriate):

Subroutine Headers:

Purpose: brief process description

CALled from: calling routine name(s)

CALLs: called routine name(s)

geometry definitions if appropriate:

On-line Flags:

Expression Comment Function

... : Default (default values set in USERx)

... : 1^4 (four-byte integer arithmetic)

III-31
SECTION SEVEN

TRANSMAP MODULE #1: RASTER TO SUBFRAME CONVERSION

7.1 Program Characteristics

Program Names: SUBFRM or SF

Subroutines: USERI
FILESI
SETUPI
SQUARE
STAGGR
LINEIN

Files: User.DAT (input user raster)
       IMAGE.DAT (output)

Task Build Options: MAXBUF = 1024

Task Size: 31936

7.2 Source Listing

The COMMENT-annotated source listing for SUBFRM follows:
### Master to Subframe Conversion: TransMAPS 1-1

#### SUBFRM

**Program Structure:**

```
CALL USERI Specify image name, position, size, partition
CALL FILES Open and position input file, open output file
CALL SETUP Characterize image partition and padding
IF square grid:
   CALL SQUARE Convert raster: 6x8 16x16 32x32 subframes
   Loop on rows of subframes in line direction
   Loop on 8-line swaths within subframe rows
       CALL LINEIN 8 calls, line input, 6-8 bit
       Loop on subframes in pixel direction
       Input prior partial subframe (if 16 32)
       Update subframe with 8 line segments
       Output updated subframe
   or IF staggered grid:
   CALL STAGGR Convert raster: 8x8 staggered subframes
   Loop on lines
       CALL LINEIN Input next line, 6-8 bit conversion
       Loop on subframes completed on this line
   Extract subframe from recirculating buffer
   Output subframe
```

**Common Block Communications:**

```
 BIU(4032,8) [Byte] Block or recirculating 8-line buffer
```

---

**Files:**

<table>
<thead>
<tr>
<th>Unit</th>
<th>Name</th>
<th>Content</th>
<th>From/To</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>In 2</td>
<td>User</td>
<td>Source raster image</td>
<td>User</td>
<td>SEGMENTED</td>
</tr>
<tr>
<td>Out 3</td>
<td>IMAGE</td>
<td>Source image, subframes</td>
<td>MAPS</td>
<td>DIRECT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>or RASTER</td>
<td></td>
</tr>
</tbody>
</table>

**User Interaction:**

- In Subroutine USERI

  - Source Image Identification
  - Source Image Position Specification
  - Source Image Size Specification
  - Source Image Partition Specification
Raster to Subframe Conversion: TransMAPS 1-2

/Header/ Standard MAPS file header

/HEADER/ SUBFRM,USERT,FILEDI,SETUPIT Length: 32

C FILE [I*2] File type
C IMA(0) [Byte] User-selected image name
C NL [I*2] Number of lines in source image
C NP [I*2] Number of pixels in source image
C NB [I*2] Number of bits/pixel in source image
C KFS [I*2] Kind of subframe 8x8 16x16 32x32
C IGRD [I*2] Subframe grid: square(0)/staggered(1)
C NS [I*4] Total subframe count
C MC [I*4] MAPSel count
C NKB [I*2] Packed block(0)/pattern(1) mode (rt-llf)
C IBV(5,2) [I*2] Optimal pattern biases by level,low/high
C IPAD(7) [I*2] Space for future extension

/IMAGIN/ Source file and position data

/IMAGIN/ USETI,FILESI,SETUPIT Length: 7

C FILNAM(10) [Byte] Source raster file name
C LSKP [I*2] Lines to skip into source raster
C KSKP [I*2] Pixels to skip into source raster

/LINEUP/ Line input control parameters

/LINEUP/ SETUPIT,SQUARE,STAGGR,LINEIN Length: 7

C IP [I*2] Initial pixel to retain (KSKP+1)
C IPR [I*2] Total input pixels to read
C LP [I*2] Last pixel with subframe completion pad
C NBT [I*2] Bit count flag 6(1) or 8(0)
C LB [I*2] Current line mod 8
C LV [I*2] Current line in input raster incl skips
C LSKPT [I*2] Lines skipped (copy of LSKP)

/SFDATA/ Subframe output data

/SFDATA/ SETUPIT,SQUARE,STAGGR Length: 512

C ISF(1024) [Byte] Subframe image assembly array

/SFTEMP/ Subframe and image size parameters

/SFTEMP/ SETUPIT,SQUARE,STAGGR,LINEIN Length: 6

C NLN [I*2] Number of lines in image
C NPT [I*2] Number of pixels in image with pad
C KSFT [I*2] Subframe size: edge
C KSBT [I*2] Subframe size: pixel count
C NLT [I*2] Subframe count

/SQR/ Square grid partition controls

/SQR/ SETUPIT,SQUARE Length: 3

C NPS [I*2] Number of pixel-direction subframes
C NLS [I*2] Number of line-direction subframes
C NSWTH [I*2] Number of swathes/subframe (1 2 4)

III-34
**Raster to Subframe Conversion: TransMAPS 1-3**

**STAG**

Staggered grid partition controls

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPSL(8)</td>
<td>(I*2) Number of staggered subframes in pixel direction for each startline mod 8</td>
</tr>
<tr>
<td>NLSL(8)</td>
<td>(I*2) Number of staggered subframes in line direction for each startline mod 8</td>
</tr>
<tr>
<td>IPSL(8)</td>
<td>(I*2) Initial pixel of first staggered subframe for each startline mod 8</td>
</tr>
</tbody>
</table>

---

```plaintext
COMMON IBUF(4096,0)
BYTE IBUF
COMMON /HEADER/ IFILE,INAME(8),NL,NP,KB,KSF,IGRD,NS,MC,NIXBP,
+ IBY(5,2),IPAL(7)
BYTE INAME
INTEGER*4 NS,MC
CALL USER!
CALL SETUP!
CALL SQUARE
GO TO 120
110 CALL STAGGR
120 CONTINUE
CLOSE(UNIT=2)
CLOSE(UNIT=3)
END
```

---

III-35
SUBROUTINE USERI

C Purpose: User interaction for raster to subframe conversion
C Source image identification
C Source image position
C Source image size
C Source image partition

C Called from: SUBFRM

COMMON /HEADER/ IFIL,INAME(N),NL,NP,NB,KS,F,IGRD,NS,MC,MIXBP,
IBV(5,2),IPAD(7)
BYTE INAME
INTEGER*4 NS,MC
COMMON /IMAGIN/ FILNAM(10),LSKP,KS
BYTE FILNAM
DIMENSION NAME(10)
BYTE NAME1,NAME1
EQUIVALENCE (NAME1,NAME1)
DATA IFIG/0/
DATA INAME*/1H/
DATA NL,NP,NB,KS,F,IGRD/460,624,8,8,0/
DATA NS,MC,MIXBP,IBV,IPAD/2#0,18#0/
DATA FILNAM/1H,1H,1HR,1HR,1HR,1H2,4#0/
DATA LSKP,KS/2#0/
DATA MPIX/4000/

C -------------------------------------

500 TYPE 500
   FORMAT(/,1X,45(1H*)),/1X,"*",/1X,43X,"*",/1X,
   "* MAPS RASTER TO SUBFRAME CONVERSION MODULE *
   
   TYPE 510
   510 FORMAT(/,3X,"SOURCE IDENTIFICATION:"
   TYPE 520,FILNAM
   520 FORMAT(/,3X,"SOURCE RASTER FILENAME (UP TO 9 CHARACTERS) ",
   TYPE 530
   1041)
   ACCEPT 1,NAMET
   1 FORMAT(100I1)
   IF(NAME1,EQ.1H ) GO TO 130
   IF(NAME1,EQ.1H/) GO TO 130
   IF(NAME1,LE.1H) AND.(NAME1,LE.1H) GO TO 110
   TYPE 530
   530 FORMAT(/,1X,"*** FILENAME MUST START WITH LETTER"
   GO TO 100
   110 DO 120 I=1,9
   FILNAM(I)=NAME1(I)
   IF(NAME1(I),LE.1H ) FILNAM(I)=0
   IF(NAME1(I),EQ.1H/) FILNAM(I)=10
   IF(I,EQ.1) GO TO 120
   IF(FILNAM(I-1),EQ.0) FILNAM(I)=0
   120 CONTINUE
Rerster to Subframe Conversion: TransMAPS 1-5

130 TYPE 540, INAME
540 FORMAT(’/5X,’USER IMAGE NAME? (UP TO 8 CHARACTERS) ’,”/A1)
ACCEPT 1,(NAME1(I),I=1,8)
DO 140 I=1,8
IF(NAME1(I),LE,1H) GO TO 150
140 CONTINUE
GO TO 170
150 IF(NAME1.EQ.,1H) GO TO 170
DO 160 I=1,8
INAME(I)=NAME1(I)
IF(INAME(I),LE,1H) INAME(I)=1H
160 CONTINUE
170 TYPE 550
550 FORMAT(/,3X,’SOURCE IMAGE POSITION’)"
TYPE 560,LSKP
560 FORMAT(/,5X,’NUMBER OF LINES TO SKIP?’,IS,5X,’(/ = NO CHNG)’)
ACCEPT *,LSKP
TYPE 570, MPX, KSKP
570 FORMAT(5X,’NUMBER OF PIXELS TO SKIP? (<”,I5,“)”,I5,4X,
’(/ = NO CHNG)’)
ACCEPT *,KSKP
IF(KSKP.GE,MPX) KSKP=MPX-1
TYPE 580
580 FORMAT(/,3X,’SOURCE IMAGE SIZE’)"
TYPE 590, NL
590 FORMAT(/,5X,’NUMBER OF LINES TO PROCESS?’,IS,5X,’(/ = NO CHNG)’)
ACCEPT *,NL
MPX=MPX-KSKP
IF(MP,GT,MPX) MP=MPX
TYPE 600,MPX, NP
600 FORMAT(5X,’NUMBER OF PIXELS TO PROCESS? (UP TO”,I5,“)”,I5,
’(/ = NO CHNG)’)"
ACCEPT *,NP
IF(NP,GT,MPX) NP=MPX
TYPE 610,NP
610 FORMAT(5X,’NUMBER OF BITS/PIXEL? (6 8)’,I3,4X,’(/ = NO CHNG)’)
IT=8
ACCEPT *,IT
IF((IT,LT,6).OR.(IT,LE,8)) NB=IT
TYPE 620
620 FORMAT(/,3X,’SOURCE IMAGE PARTITION’)"
TYPE 630,KSF
630 FORMAT(/,5X,’SUBFRAME EDGE? (0 16 32)’,I3,7X,’(/ = NO CHNG)’)
IT=KSF
ACCEPT *,IT
IF((IT,LT,8).OR.(IT,LE,16).OR.(IT,LE,32)) KSF=IT
IF(KSF,NE,0) GO TO 180

III-37
LGRD=1HN
IF(IGRD .NE. 0) LGRD=1HY
TYPE 640, LGRD
640 FORMAT(5X, "STAGGER GRID? (Y OR N) ", A1)
ACCEPT 1, LIT
IF(LIT.EQ. '1'H) IGRD=0
IF(LIT.EQ. '1'HY) IGRD=1
GO TO 190
180 IGRD=0
190 TYPE 650
650 FORMAT(//, 3X, "USER SPECIFICATION COMPLETE", //, 3X, 28('1HN'), //, 5X,
  "REVIEW? (Y OR N) ")
ACCEPT 1, LIT
IF(LIT.EQ. '1'HY) GO TO 100
RETURN
END
SUBROUTINE FILES!

Purpose: Open and position files

Called from: SUBFRM

COMMON /NEADER/ iriLEINANE(8),ML,MP,NB,KSF,IGRD,NS,MC,MIXBP,
+ 1BV(5,2),IPAD(7)
BYTE INAME
INTEGER*4 NS,MC
COMMON /INAGIN/ FILNAM(IO),LSKP,K$KP
BYTE FILNAM
OPEN(UNIT=2,TYPE="OLD",NAME=FILNAM,FORM="UNFORMATTED",READONLY)
IF(LSKP.LE.0) GO TO 130
DO 110 L=1,LSKP
READ (2,END=120,ERR=120)
110 CONTINUE
GO TO 130
120 TYPE 500,L
500 FORMAT(/,IX,"*** EOF/ERR AT SKIP LINE",IS)
STOP
130 CONTINUE
LSF=(KSF*KSF)/4
OPEN(UNIT=3,TYPE="NEW",NAME=IMAGE,FROM="UNFORMATTED",
+ RECORDTYPE="FIXED",RECORD=SIZE=LSF,ACCESS="DIRECT")
RETURN
END
SUBROUTINE SETUPI

C Purpose: Establish subframe partition parameters
C Write subframe output file standard MAPS header
C
C CALLED from: SUBFRM
C
COMMON /HEADER/ IFILE, INAME(8), NL, NP, NB, KSF, IGRD, MS, MC, MIXBP,
+ IBV(5,2), IPAD(7)
BYTE INAME
INTEGER*4 NS, MC
DIMENSION JHEAD(32)
EQUIVALENCE (JHEAD(1), IFILE)
COMMON /IMAGIN/ FILNAM(10), LBKP, KSKP
BYTE FILNAM
COMMON /LINEUP/ IP, IPR, LP, NBT, LB, LK, LKPT
COMMON /SQK/ MPS, NLS, NS=TH
COMMON /STG/ MPSL(5), LSL(5), IPSL(5)
DIMENSION JPSL(5), JLSL(5)
COMMON /SFTEMP/ NLT, NPT, KSFT, KSOT, NST
INTEGER*4 NST
COMMON /SFDATA/ ISF(1024)
BYTE ISF
DIMENSION JSF(512)
EQUIVALENCE (JSF(1), ISF(1))
INTEGER*4 1STAR4
DATA IPSL/1, 25, 49, 9, 33, 57, 17, 41/
DATA JPSL/7, 4, 1, 6, 3, 0, 5, 2/, JLSL/7, 14, 13, 12, 11, 10, 9, 8/
DATA JSF/512*0/
NPS=(NP+1)/KSF+1
IP=KSKP+1
IPR=KSKP+NP
LP=KSKP+KSFT*NP
NBT=0
IF(NB.EQ.6) NBT=1
LSKPT=LSKP
IF(IGRD.NE.0) GO TO 110
C
C Square Grid Partition
C
WLS=(NL-1)/KSF+1
1STAR4=NPS
NS=1STAR4*NLS
NSWTH=1
IF(KSF.EQ.16) NSWTH=2
IF(KSF.EQ.32) NSWTH=4
GO TO 130
110 CONTINUE
Staggered Grid Partition

NS=0
DO 120 J=1,8
NPSL(J)=(NPS+JPSL(J))/8
ISTAR4=NPSL(J)
NLSL(J)=(NL+JLSL(J))/8
NS=NS+NLSL(J)*ISTAR4
120 CONTINUE
130 NL=NL
HPT=KSF*PS
KSF=KSF
KSQT=KSF*KSF
MST=NS
DO 140 J=1,32
140 JBF(J)=JHEAD(J)
WRITE (31) (1SF(J), J=1, KSQT)
RETURN
END
SUBROUTINE SQUARE

Purpose: Convert source image raster to square grid of subframes

Loop on rows of subframes in line direction
Input next 8 lines
Loop on 8-line swaths within subframe rows
Input prior partially-completed subframe
Update subframe with 8 line segments
Output updated subframe

Called from: SUBFRM

Calls: LINEIN

COMMON IBUF(4032,8)
BYTE IBUF
COMMON /LINEUP/ IP,IPK,LP,NBT,LG,LS,LKPT
COMMON /SGN/ NPS,NLS,NSWTH
COMMON /SFTEMP/ NLT,NPT,KSFT,KSQT,NST
INTEGER*4 MST
COMMON /SFDATA/ ISF(1024)
BYTE ISF
INTEGER*4 KREC,JREC
L=0
KREC=1

DO 300 JLS=1,NLS
   DO 270 JSWTH=1,NSWTH
      DO 140 JLIN=1,8
         L=L+1
         LK=L+LKPT
         IF(L.LE,NLT) GO TO 120
         JPREV=JLIN-1
         IF(JPREV.EQO) JPREV=8
         DO 110 JPIX=1,NPT
            IBUF(JPIX,JLIN)=IBUF(JPIX,JPREV)
            GO TO 140
      110
      JPREV=JLIN
      CONTINUE
      JREC=KREC
      JIP=1
      JLP=KSFT
      JSQT=JSOT#(JSWTH=1)
      DO 260 JPS=1,NPS
         JREC=JREC+1
      260 JSQT=JSOT+1
      IF(JSWTH.NE.1) GO TO 220
      DO 210 JSQT=1,KSQT
         ISF(JSQT)=0
         GO TO 230
      210
      READ (3,JREC) (ISF(JSQT),JSQT=1,KSQT)

III-42
Raster to Subframe Conversion: TransMAPS 1-11

SUBFMR/SQUARE

230    JSQT=ISQT
   DO 250 JLIN=1,8
   DO 240 JPIX=JIP,JLP
      JSQT=JSQT+1
   240    ISF(JSQT)=IBUF(JPIX,JLIN)
   250    CONTINUE
   &RITE (J'JREC) (ISF(JSQT),JSQT=1,KSQT)
      JIP=JIP+KSFT
   260    JLP=JLP+KSFT
   270    CONTINUE
   300    KREC=KREC+1
   RETURN
   END
SUBROUTINE STAGGR

C

C Purposes: Convert source image raster to staggered grid of subframes

Loop on lines
Input next line
Loop on subframes completed on this line
Extract subframe from recirculating buffer
Output subframe

Called from: SUBFRM

Calls: LINEIN

C-------------------------------............

COMMON IBUF(4032,8)
BYTE IBUF
COMON /LINEUP/ IP,IPR,LP,MST,L8,LK,LSKPT
COMMON /STGR/ NPSL(8),NLSS(8),JPSL(8)
COMMON /SFTEMP/ MTL,NPT,KSFT,KSOT,MST
INTEGER*4 MST
COMMON /SFDATA/ ISF(1024)
BYTE ISF
INTEGER*4 JREC
JREC=1
NL7=NL7+7
DO 300 L=1,NL7
LK=L+LSKPT
LSTRT=L AND *7
LS1=LSTRT+1
IF(LSTRT.EQ.0) LSTRT=8
IF(L.LE.NL7) GO TO 120
JPREV=LSTRT-1
IF(JPREV.EQ.0) JPREV=8
DO 110 JPIX=1,NPT
110 IBUF(JPIX,LSTRT)=IBUF(JPIX,JPREV)
GO TO 140
120 L0=LSTRT
CALL LINEIN
140 IF(L.LT.1) GO TO 200
DO 160 JLIN=2,8
DO 150 JPIX=1,NPT
150 IBUF(JPIX,JLIN)=IBUF(JPIX,1)
160 CONTINUE
200 CONTINUE
JPSL=NPSL(LS1)
IF(JPSL.EQ.0) GO TO 300
JIP=IPSL(LS1)
JLP=JIP+7

III-44
DO 230 JPS=1, JPLS
  JLIN=LS1
  JSQT=0
  DO 220 JLT=1,0
    DO 210 JPIX=JIP, JLP
    JSQT=JSQT+1
    ISF(JSQT)=IBUF(JPIX, JLIN)
  210
    JLIN=JLIN+1
    IF(JLIN, GT, 0) JLIN=1
  220
    CONTINUE
    JREC=JREC+1
    WRITE (3, JREC) (ISF(JSQT), JSQT=1, KSQT)
    JIP=JIP+64
    JLP=JLP+64
  230
  300 CONTINUE
RETURN
END
SUBROUTINE LINEIN

C Purpose: Input next line from source image raster input file
C Pad line to integral number of subframes
C Convert pixels from 6 bits to 8 bits if designated
C Skip designated number of input pixels
C
C Called from: SQUARE, STAGGR
C
COMMON IBUF(4032,0)
BYTE IBUF
COMMON /LINEUP/ IP,IPR,LP,NBT,LB,LK,LSKPT
COMMON /SFTENP/ NLT,NPT,KSFT,KSQT,NST
INTEGER*4 NST
DIMENSION KPIX(64)
DATA KPIX/4,10,14,20,24,30,34,40,44,50,54,60,64,
  70,74,100,104,110,114,120,124,130,134,140,144,
  150,154,160,164,170,174,200,204,210,214,220,224,
  230,234,240,244,250,254,260,264,270,274,300,304,
  310,314,320,324,330,334,340,344,350,354,360,364,
  370,374/
READ (2,END=110,ERR=110) (IBUF(J,LB),J=1,IPR)
GO TO 120
110 TYPE 500,LK,LSKPT
500 FORMAT(/1X,'*** EOF/ERR AT LINE'15,' (INCLUDING', IS,
  'SKIPS')
STOP
120 DO 130 J=IPR,LP
130 IBUF(J+1,LB)=IBUF(J,LB)
IPIX=IP
DD 140 JPIX=1,NPT
KPIX=IBUF(IPIX,LB)
IPIX=IPIX+1
IF(NBT,EQ.0) GO TO 140
KP6=KPIX.AND.77
KPIX=KPIX6(KP6+1)
140 IBUF(JPIX,LB)=KPIX
150 RETURN
END
SECTION EIGHT

TRANSMAPS MODULE #2: MAPS COMPRESSION

8.1 Program Characteristics

Program Names: MAPS or MP

Subroutines: FILESM
            USERM
            SETKCM
            SETURM
            RTPSET
            ZIGZAG
            LVLSET
            SFMAC
            SFIN
            SFMAPS
            QDIFF
            THRESH
            MAPOUT
            LSTREC
            SUMMARY

Files: MSET.DAT (input/output)
       IMAGE.DAT (input)
       MAPS.DAT (output)

Task Build Options: MAXBUF = 1024

Task Size: 31488

8.2 Source Listing

The COMMENT-annotated source listing for MAPS follows:
MAPS Compression: TransMAPS 2-1

PROGRAM MAPS

| TransMAPS Module #2: MAPS Compression |

Control Data Corporation - 1982

Files: Unit Name Content From/To Type

In/Out 1 MSET User-defined parameter set MAPS SEGMENTED
In 2 IMAGE Source image, subframes SUBFRM FIXED SEQ.
Out 3 MAPS MAPSel stream, 512 byte rec. DMAPS DIRECT

User Interaction: In Subroutine USERM

Macro-fidelity Control
Micro-fidelity Control
Gray-scale Manipulations

Program Structure:

Program MAPS

CALL FILES4 Open files, read file header for source image
CALL USERM Specify macro-fidelity, micro-fidelity, and gray-scale controls
CALL SETKCM(K) Establish contrast control matrix, Set K
CALL SETUPM Establish compression control tables
CALL ZIGZAG Establish raster to zigzag reorder table
CALL LVSET Establish 3 MAPSel/byte level packing
Loop on subframes (LOOPSF)
CALL SFMAC(LOOPSF) Determine macro-fidelity position
CALL SFIN Input subframe, map gray scales, zigzag order
CALL SFMAPS MAPS subframe compression kernel
Loop on levels (resolution)
Loop on MAPSel quads
CALL ODIF1 Form contrasts, sign-sort vector
CALL THRESH Test contrasts, form MAPSel
Transfer completed MAPSels to output buffer
CALL MAPOUT Output MAPSel stream record
CALL LSTREC Output final (partial) MAPSel stream record
CALL SUMMRX Determine optimal biases, compression, & MSE

COMMON Block Communication:

/BLKPAT/ Block/pattern mode by level Length: 21
SETUPM, SFRMAPS

LBPT(5) [I*2] Block(0)/pattern(1) for levels 1-5
IQBP(4,4) [I*2] Low/High index by Quadrant, Pattern
MAPS Compression: TransMAPS 2-2

/ CNTRST/ Functional contrast control matrix
  Length: 81
  SETUPM,SFMAC,SFMAPS,THRESH

  KCM(80)  [I*2]  Sequentially-addressed contrast matrix
  KINDEX  [I*2]  Pointer for current macro-fidelity index

/CTRL/ User-interactive input specifications
  Length: 415
  USERM,SETKCM,SETUPM,SRPSET,SRMAC

  MAC(16,16)  [I*2]  Macro-fidelity image partition
  KCS(4)  [I*2]  Micro-type: parametric(0)/matrix(1)
  CS(4)  [R*4]  Contrast scale parameter
  TB(4)  [R*4]  Taper base parameter
  SF(4)  [R*4]  Step fraction parameter
  SB(4)  [R*4]  Step bias parameter
  KCM(4,5,4)  [I*2]  Contrast control matrices
    (Contrast,Transition,Macro-group)

  LBP(6)  [I*2]  Block(0)/pattern(1) mode (level+i)
  KSP(2,9)  [I*2]  Contrast-space remap breakpoint pairs
  IBP(2,9)  [I*2]  Intensity-space remap breakpoint pairs
  IRSET  [I*2]  Intensity reset type (M P L S T H)

/GRYSCL/ Gray-scale remap tables
  Length: 4608
  SETUPM,RMPSET,SRF,SFMAPS,THRESH

  KRMPC25S  [I*2]  Code to Contrast space remap
  IRMP(256)  [I*2]  Code to Intensity space remap
  IDMP(4096)  [I*2]  Intensity to Code space demap

/HEADER/ Standard MAPS file header
  Length: 32
  MAPS,FILE,USERM,SETUPM,SUMMARY

  FILE  [I*2]  File type
  INAME(8)  [Byte]  User-selected image name
  ML  [I*2]  Number of lines in source image
  NP  [I*2]  Number of pixels in source image
  MB  [I*2]  Number of bits/pixel in source image
  KSF  [I*2]  Kind of subframe 8x8 16x16 32x32
  IGRD  [I*2]  Subframe grid: square(0)/stepped(1)
  NS  [I*4]  Total subframe count
  MC  [I*4]  MAPSel count
  MIXBP  [I*2]  Packed block(0)/pattern(1) mode (rt-lft)
  IBV(5,2)  [I*2]  Optimal pattern biases by level,low/high
  IPAD(7)  [I*2]  Space for future extension

/LVLTBL/ Resolution (level) code packing table
  Length: 366
  SETUP,MVSET,MAPOUT,LSTREC

  LVLT(366)  [I*2]  Level code triplet to byte conversion

III-49
MAPS Compression: TransMAPS 2-3

/ MAPSSF / MAPS Subframe data spaces

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSQ</td>
<td>Number of total pixels/subframe</td>
<td></td>
</tr>
<tr>
<td>NLVL</td>
<td>Number of active levels</td>
<td></td>
</tr>
<tr>
<td>ICODE(1024)</td>
<td>Subframe pixels in code (8 bit) space</td>
<td></td>
</tr>
<tr>
<td>KNTRST(1024)</td>
<td>Pixel/MAPSel map to contrast space</td>
<td></td>
</tr>
<tr>
<td>INTENS(1024)</td>
<td>Pixel/MAPSel map to intensity space</td>
<td></td>
</tr>
<tr>
<td>LEVEL(1024)</td>
<td>MAPSsel resolution (level) code</td>
<td></td>
</tr>
<tr>
<td>PATTERN(1024)</td>
<td>MAPSsel pattern code</td>
<td></td>
</tr>
</tbody>
</table>

/ MDATA / MAPS output buffer

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>MREC</td>
<td>Index of most recently written record</td>
<td></td>
</tr>
<tr>
<td>NLOC</td>
<td>Index of most recent buffer entry</td>
<td></td>
</tr>
<tr>
<td>MS(512)</td>
<td>MAPSsel stream assembly buffer</td>
<td></td>
</tr>
</tbody>
</table>

/ MSTATS / Accumulators for optimum bias & MSE

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>KOUNT(2,b)</td>
<td>Pixel count by low/high,level</td>
<td></td>
</tr>
<tr>
<td>DIFF(2,6)</td>
<td>Sum of (I-N) by low/high,level</td>
<td></td>
</tr>
<tr>
<td>DIFFSQ</td>
<td>Sum square of (I-N) [source-MAPS]</td>
<td></td>
</tr>
<tr>
<td>SUMSQ</td>
<td>Sum square of 1 [source intensities]</td>
<td></td>
</tr>
</tbody>
</table>

/ MTEMP / Temporary staging, partial multiplets

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM</td>
<td>Multiplet size (4 for 8x8, 3 for 16, 32)</td>
<td></td>
</tr>
<tr>
<td>NW</td>
<td>Count currently in list</td>
<td></td>
</tr>
<tr>
<td>MAPSEL(5)</td>
<td>MAPS intensity/pattern values</td>
<td></td>
</tr>
<tr>
<td>NLVL(4)</td>
<td>MAPS resolution (level) codes</td>
<td></td>
</tr>
</tbody>
</table>

/ QUAD / Current quad of MAPS elements

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>KT(4)</td>
<td>Contrast space quad</td>
<td></td>
</tr>
<tr>
<td>IT(4)</td>
<td>Intensity space quad</td>
<td></td>
</tr>
<tr>
<td>NTV(6)</td>
<td>Contrasts: 3-2, 3-1, 2-0, 1-0, 2-1, 3-0</td>
<td></td>
</tr>
<tr>
<td>LG</td>
<td>Location of quad (zigzag index of start)</td>
<td></td>
</tr>
<tr>
<td>LVLP</td>
<td>Level resulting if quad is combined</td>
<td></td>
</tr>
<tr>
<td>NW</td>
<td>Contrast-sign sort vector</td>
<td></td>
</tr>
</tbody>
</table>

/ RESET / Intensity reset controls

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSK(4)</td>
<td>Activation masks by sort order</td>
<td></td>
</tr>
<tr>
<td>NORM</td>
<td>Normalization divisor (4 2 1)</td>
<td></td>
</tr>
<tr>
<td>NBIAS</td>
<td>Bias for rounding (2 1 0)</td>
<td></td>
</tr>
</tbody>
</table>

III-50
NAPS Compression: TransNAPS 2-4

**/SFCNTL/**
- **KSSFT** ([*2]) Subframe size: Edge (0 16 32)
- **KSDQ** ([*2]) Subframe size: Pixel count
- **MPST** ([*2]) Number of subframes in pixel direction
- **NPSL(8)** ([*2]) Number of staggered subframes in pixel direction for each startline mod 8
- **NPSL(8)** ([*2]) Middle pixel of first staggered subframe for each startline mod 8
- **KSFH** ([*2]) Subframe half size
- **FL** ([*4]) Line to Macro-fidelity index factor
- **FP** ([*4]) Pixel to Macro-fidelity index factor
- **JGRD** ([*2]) Grid type: square(0)/staggered(1)

**/SFDATA/**
- **ISF(1024)** [Byte] Subframe input array

**/ZIGZAG/**
- **IZZ(1024)** ([*2]) Raster to zigzag lookup table

Zigzag Order Convention:
```
+-----+-----+-----+-----+
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Line direction</td>
<td>MAPSel Quadrants</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td>V</td>
</tr>
</tbody>
</table>

+-----+-----+-----+-----+
```

COMMON /HEADER/ IFILE,INAME(*),NL,MP,MN,KSF,JGRD,NS,MC,MIXBP,
IBV(5,2),IPAD(7)
BYTE INAME
INTEGER NS,MC
DIMENSION JHEAD(32)
EQUIVALENCE (JHEAD(1),IFILE)
COMMON /DATA/ MREC,NLOC,NSF(512)
BYTE NSF
INTEGER MREC
DIMENSION JSF(256)
EQUIVALENCE (JSF(1),NSF(1))
INTEGER LOOPSF
CALL FILESM
CALL USERM

III-51
MAPS Compression: TransMAPS 2-5

TYPE 500,INAME,NL,NP
FORMAT(1X,'MAPS COMRESSING IMAGE ",BA1","IS," LINES BY", I5," PIXELS")
CALL SETUPN
DO 110 LOOPSF=1,NS
CALL SFMAC(LOOPSF)
CALL SFIM
CALL SFMAPS
110 CONTINUE
CALL LISTREC
CALL SUMMR
DO 120 J=1,256
120 JSF(J)=0
DO 130 J=1,32
130 JSF(J)=JHEAD(J)
WRITE (3') JSF
CLOSE(UNIT=2)
CLOSE(UNIT=3)
END
SUBROUTINE FILESN

C Purpose: Open files, read input header, write preliminary output header

C CALLED from: MAPS

COMMON /HEADER/ IFIL£E,INAME(8),ML,MP,MB,KSF,GRID,NS,MC,MIXBP,
+ IBV(5,2),IPAD(7)
BYTE INAME
INTEGER*4 MC
DIMENSION JHEAD(32)
EQUIVALENCE (JHEAD(1),IFILE)
COMMON /MDATA/ MREC,MSOC,MSF(512)
BYTE MSF
INTEGER*4 MREC
DIMENSION JSF(256)
EQUIVALENCE (JSF(1),MSF(1))
OPEN(UNIT=2,TYP£8'OLD',NAME="IMAGE",FORM="UNFORMATT£D",
+ RECORDTYPE="FIXED")
READ (2) JHEAD
OPEN(UNIT=3,TYP£8'NEW',NAME="MAP£S",FORM="UNFORMATT£D",
+ RECORDTYPE="FIXED",RECORDSIZE=128,ACCESS="DIRECT")
DO 110 J=1,256
110 JSF(J)=0
DO 120 J=1,32
120 JSF(J)=JHEAD(J)
WRITE (3'1) JSF
RETURN
END
SUBROUTINE USERM

C Purposes: User interaction for MAPS compression
C Mode: Quick User Full
C MAPS macro-fidelity control
C MAPS micro-fidelity control
C MAPS gray-scale manipulations

C CALLS: SETKCM

C---------------------------------------------------------------------
COMMON /CTRL/ MAC(16,16),KCS(4),CS(4),TB(4),SF(4),SB(4),
+ KCS(5,4),LBP(6),KBP(2,9),IBP(2,9),IRSET
COMMON /HEADER/ IFILEINAME,U),NL,NP,NB,KSrIGRD.NS,NC,NMIXBP,
+ IBV(5,2),IPAD(7)
DIMENSION MSET(415)
EQUIVALENCE (MSET,MAC)
BYTE INAME
INTEGER*4 NS,MC
DIMENSION AT(lb),IAC(4)
DATA CSD,ThDSFu.SBD#/2O.O.3.O,0.5,0.1/ ! Default
DATA LBP2,L8P3.LBP4,LBP5.LBP6 u255
DATA MAXK,MAMI/4095,4095/
DO 10 L=1,16
DO 10 I=1,16
MAC(I,L)=1 ! Default
DO 20 K=1,4
KCS(K)=1HP
CS(K)=CSD
TB(K)=TBD
SF(K)=SFD
SB(K)=SBD
CALL SETKCM(K)
20 CONTINUE
LBP(1)=0
LBP(2)=LBP2
LBP(3)=LBP3
LBP(4)=LBP4
LBP(5)=LBP5
LBP(6)=LBP6
KBP(1,1)=0
KBP(2,1)=0
IBP(1,1)=0
IBP(2,1)=0
DO 30 J=2,9
KBP(1,J)=255 ! Default
KBP(2,J)=255 ! Default
IBP(1,J)=255 ! Default
IBP(2,J)=255 ! Default
30 CONTINUE
IRSET=1NM

III-54
NAPS Compression: TransMaps 2-0

TYPE 500

500 FORMAT(/,1X,27(1H4),/,,1X,"*",25X,"*",/,,1X,
+ "# MAPS COMPRESSION MODULE #",
+ /,,1X,"*",25X,"*",/,,1X,27(1H4))

TYPE 510

510 FORMAT(/,3X,"USER OPTION MODE?"),
+ //,5X,"Q - QUICK MODE (SELECT CONTRAST SCALE ONLY)",
+ //,5X,"U - USE USER PRE-DEFINED PARAMETERS FROM FILE MSET.DAT",
+ //,5X,"F - FULL OPTION REVIEW AND SELECTIVE REVISION",
+ //,5X,"MODE? (O U F) Q")

ACCEPT 1,LIT

1 FORMAT(10A1)

IF(LIT.EQ.1H4) GO TO 100
IF(LIT.EQ.1H4) GO TO 80

TYPE 520,CS(1)

520 FORMAT(/,5X,"CONTRAST SCALE?",F7.1,5X,"(/ = NO CHNG")

ACCEPT *,CS(1)

CALL SETKCN(1)

GO TO 400

80 OPEN(UNIT=1,TYPE="OLD",NAME="MSET",FORM="UNFORMATTED",ERR=90)

READ (1) MSET

CLOSE(UNIT=1)

GO TO 400

90 TYPE 530

530 FORMAT(/,1X,"## NO PRE-DEFINED PARAMETER FILE FOUND!,",
+ "SET DIRECTLY")

C

MAPS Macro-fidelity Control

C

100 TYPE 540

540 FORMAT(/,3X,"MACRO-FIDELITY CONTROL: REVIEW/REVISE? ",
+ "(Y ON N)")

ACCEPT 1,LIT

IF(LIT.NE.1H4) GO TO 140

TYPE 550,MAC(1,L),1=1,16,L=1,16)

550 FORMAT(/,12X,"CURRENT IMAGE PARTITION",/,
+ 16(//,8X,25X,"ROW",/,,5X,"ROW TO CHANGE? (1-16)",
+ 5X,"(/ = NO FURTHER CHNG")

L=0

ACCEPT *,L

IF((L.LT.1).OR.(L.GT.16)) GO TO 140

DD 120 L=1,16

120 MT(1)=MAC(1,L)

TYPE 560,L,MT

560 FORMAT(3X,"REVISE ROW",12,"? (RANGE: 1-4) 
+ (/ = NO CHNG)",
+ ,16(1X,11))

ACCEPT *,MT

DD 130 L=1,16

IT=MT(1)

IF((IT.GE.1).AND.(IT.LE.4)) MAC(1,L)=IT

130 CONTINUE

GO TO 110

140 DD 150 K=1,4

150 IA(K)=0

DD 160 L=1,16

DD 160 L=1,16

IT=MAC(1,L)

160 IA(IT)=IA(IT)+1

III-55
MAPS Compression: TransMAPS 2-9

C *** MAPS Micro-fidelity Control

C TYPE 570

570 FORMAT(/,3X,"MICRO-FIDELITY CONTROL: REVIEW/REVISE? ",
+ "(Y OR N) N")
ACCEPT 1.,LIT
IF(LIT.NE.1.)GO TO 300
DO 270 K=1,4
IF(IAK(K).LE.U) GO TO 270
IF(KC5(K).EQ.1.)GO TO 220

210 TYPE 580,K,CS(K),TB(K),SF(K),SB(K)

220 FORMAT(/,7X,"GROUP",12," CONTRAST THRESHOLD PARAMETERS",
+ "/.9X,"CONTRAST SCALE",F7.1,"/9X,"TAPER",9X,F7.1,
+ "/.9X,"STEP FRACTION",F7.1,"/9X,"STEP BIAS ",F7.1)

230 TYPE 590,K,(L-1,L,(KCM(J,L,K),J=1,4),L,L=1,LT)

590 FORMAT(/,7X,"GROUP",12," CONTRAST THRESHOLD MATRIX",
+ "/.16X,"/E M L U",5(/,9X,11,"="E14.5,51,1,ROW",12))

600 FORMAT(/,5X,"SPECIFICATION MODES!",/7X,"M - NO CHANGE",
+ "/.7X,"S - SCALE ONLY",/7X,"P - PARAMETRIC",/7X,"N - MATRIX",
+ "/.7X,"REVISE SPECIFICATIONS? (N S P M) N")
ACCEPT 1.,LIT
IF(LIT.NE.1.)GO TO 250

240 TYPE 600

KCS(K)=1

610 FORMAT(/,5X,"MATRIX ROW TO CHANGE? (1="E11,),5X,
+ "(= NO FURTHER CHNG)"
ACCEPT *,L
IF((L,LT.1).OR.(L.GT.LT)) GO TO 270

620 TYPE 620,K,L-1,L,(KCM(J,L,K),J=1,4)

+ "/.4X,"/E M L U",5(/,9X,11,"="E14.5,51,1,NO CHNG)"
ACCEPT *,(KCM(J,L,K),J=1,4)

TYPE 590,K,(L-1,L,(KCM(J,L,K),J=1,4),L,L=1,LT)
GO TO 240

630 IF(LIT.NE.15).AND.(LIT.NE.1.)GO TO 270
KCS(K)=1

640 TYPE 610

KCS(K)=1

650 FORMAT(5X,"GROUP",12," CONTRAST SCALE?",F7.1,5X,
+ "/= NO CHNG)"
ACCEPT *,CS(K)
IF(LIT.EQ.1.)GO TO 260

660 TYPE 640,K,TB(K)

670 IF(LIT.0EQ.1.)GO TO 260

680 TYPE 650,K,SF(K)

690 IF(LIT.0EQ.1.)GO TO 260

700 CALL SETKCM(K)
GO TO 210

270 CONTINUE
MAPS Compression: Transmission

DO 280 L=1,LVL
MT(L)=1HB
IF (LBP(L),NE.0) MT(L)=1HP
280 CONTINUE

TYPE 670,(L-1,L1,LVL)
670 FORMAT(/,3X,"BLOCK/PATTERN ASSIGNMENT: "//,9X,"LEVEL ",6I1)
TYPE 680,(MT(L),L=1,LVL)
680 FORMAT(9X,"MODE ",6A1)
TYPE 690,(MT(L),L=1,LVL)
690 FORMAT(/,5X,"REVISE B/P VECTOR? ",6A1)

ACCEPT 1,(MT(L),L=1,LVL)
DO 290 L=2,LVL
IF (MT(L).EQ.1HB) LBP(L)=0
IF (MT(L).EQ.1HP) LBP(L)=1
290 CONTINUE

C *** MAPS Gray-scale Manipulations

C

300 TYPE 700
700 FORMAT(/,3X,"GRAY-SCALE MANIPULATIONS: REVIEW/REVISE? ",
+ "(Y OR N) ")
ACCEPT 1,LIT
IF (LIT.NE.1HY) GO TO 400
310 DO 320 K=2,9
IF (KBP(1,K).GE.255) GO TO 330
320 CONTINUE
330 K=9
340 TYPE 710,(KBP(1,J),KBP(2,J),J=1,K)
710 FORMAT(/,5X,"CONTRAST SPACE REMAPPING: PIECEWISE LINEAR ",//,
+ 7X,"(CODE SPACE/CONTRAST SPACE) BREAKPOINT PAIRS ",//,
+ (14,13,6X,14))

TYPE 720
720 FORMAT(/,5X,"REVISE CONTRAST REMAP? (Y OR N) ")
ACCEPT 1,LIT
IF (LIT.NE.1HY) GO TO 350
DO 340 K=2,9
340 CONTINUE

C *** MAPS/USERN

III-57
NAPS Compression: TransNAPS 2-11

GO TO 310
350 DO 360 I=2,9
   IF(IBP(1,J).GE.255) GO TO 370
360 CONTINUE
   I=9
370 TYPE 740,(IBP(1,J),IBP(2,J),J=1,1)
740 FORMAT(/,5X,'INTENSITY SPACE REMAPPING: PIECEWISE LINEAR',//,
   + '7X,'(CODE SPACE/INTENSITY SPACE) BREAKPOINT PAIRS',//,
   + '(1I1,13,6X,14)')
TYPE 750
750 FORMAT(/,5X,'REVISE INTENSITY REMAP? (Y OR N) N')
   ACCEPT 1,LIT
   IF(LIT.NE.1MH) GO TO 390
   DO 380 I=2,9
      II=IBP(1,I)
      IF(I.EQ.9) I1L=255
      I2L=IBP(2,I)
      I1U=255
      I2U=MAXI
      I1T=IBP(1,I)
      IF(I1T.LE.II1) I1T=I1L
      I2T=IBP(2,I)
      IF(I2T.LE.I2L) I2T=I2L
      TYPE 760,1,111,11UL,12L,12U,1,11T,12T
    760 FORMAT(/,5X,'POINT',12,'/',9X,'CODE SPACE RANGE',7X,
      + '7X,'(13,=','13,')',/9X,'INTENSITY SPACE RANGE (',14,=','14,
      + ')',/5X,'REVISE',12,' (CODE/INTENSITY)?',14,IS,
      + '(/ = NO ChNG)')
   ACCEPT 1.11T,12T
      IF(I1T.LE.I1L) I1T=I1L
      IF(I1T.GT.I1U) I1T=I1U
      IBP(1,I)=I1T
      IF(I2T.LE.I2L) I2T=I2L
      IF(I2T.GT.I2U) I2T=I2U
      IBP(2,I)=I2T
      IF(LIT.EQ.255) GO TO 350
   CONTINUE
380 CONTINUE
GO TO 350
390 TYPE 770,LRESET
770 FORMAT(/,5X,'INTENSITY RESET?',//,7X,'M = MEAN OF QUAD',//,7X,
      + 'P = PSEUDO-MEAN OF QUAD',//,7X,'L = LOWEST IN QUAD',//,7X,
      + 'S = SECOND IN QUAD',//,7X,'T = THIRD IN QUAD',//,7X,
      + 'H = HIGHEST IN QUAD',//,5X,'REVISE RESET? (N P L S T M) ?',
      + 'AL')
   ACCEPT 1,LIT
      IF(LIT.EQ.1HM) LRESET=1HM
      IF(LIT.EQ.1HP) LRESET=1HP
      IF(LIT.EQ.1HL) LRESET=1HL
      IF(LIT.EQ.1HS) LRESET=1HS
      IF(LIT.EQ.1HT) LRESET=1HT
      IF(LIT.EQ.1HM) LRESET=1HM

III-58
400 TYPE 780
780 FORMAT(/,3X,"USER SPECIFICATION COMPLETE:",/,,3X,28(1X),/,,5X,
     + "REVIEW? (Y OR N) N")
       ACCEPT 1,LIT
       IF(LIT.EQ.1N) GO TO 100
       TYPE 790
790 FORMAT(3X,"SAVE THESE PARAMETERS FOR FUTURE USE? (Y OR N) N")
       ACCEPT 1,LIT
       IF(LIT.NE.1N) GO TO 410
       OPEN(UNIT=1,TYPE="NEW",NAME="MSET",FORM="UNFORMATTED")
       WRITE (1) MSET
       ENDFILE 1
       CLOSE(UNIT=1)
       TYPE 800
800 FORMAT(5X,"PARAMETERS SAVED ON FILE MSET.DAT")
410 CONTINUE
       RETURN
       END
SUBROUTINE SETKCM(K)

C Purpose: Inter Group K contrast control matrix from parametric specification

C CALLED from: USERM

COMMON /CONTRL/ MAC(16,16),KCS(4),CS(4),TB(4),SF(4),SB(4),
+ KCM(4,5,4),LbP(6),KBP(2,9),IBP(2,9),IRSET

TE=CS(K)
B=TB(K)
FM=SF(K)
PD=FM+SB(K)
IF(B.GT.0.) GO TO 10
TYPE 100,B,K

100 FORMAT(/** GROUP',12,' TAPER =''',F8.1)

10 DO 20 L=1,5
   IT=TE*0.5
   KCM(1,L,K)=IT
   IT=FM*TE*0.5
   KCM(2,L,K)=IT
   IT=FM*TE*0.5
   KCM(3,L,K)=IT
   KCM(4,L,K)=IT
20   TE=TE/B
RETURN
END
SUBROUTINE SETUPM

C
C Purposes Establish MAPS compression controls
C
C Pack block(0)/pattern(1) modes to bit vector by level
C
C Establish subframe partition parameters for macro-fidelity
C
C index determination
C
C Establish contrast control matrices with single index
C
C addressing
C
C Establish contrast remap and intensity remap/demap tables
C
C Establish subframe raster to zigzag conversion
C
C Establish level-triplet to byte resolution packing table
C
C Initialize optimum bias and performance evaluation
C
C accumulators

C CALLED from: MAPS
C
C CALLS: RNPSETZIGZAG,LVLSET

COMMON /CONTRL/ MAC(16,16),KCS(4),CS(4),TB(4),SF(4),SB(4),
       KCH(4,5,4),LBP(6),KB(2,9),LB(2,9),IRSET
COMMON /HEADER/ IFILE,INAME(8),NL,NP,M5,KSF,IGRD,NS,NC,MIXBP,
       IBV(5,2),IPAD(7)
BYTE INAME
INTEGER NS,NC
COMMON /ICONTRL/ MACC16,16),KCS(4),CS(4),TB(4),SF(4),SB(4),
       KCH(4,5,4),LBP(6),KB(2,9),LB(2,9),IRSET
COMMON /HEADER/ IFILE,INAME(8),NL,NP,M5,KSF,IGRD,NS,NC,MIXBP,
       IBV(5,2),IPAD(7)
BYTE INAME
INTEGER NS,NC

COMMON /SFCNTL/ KSFT,KSF,T,NPSL(8),NPSL(8),KSFH,FL,FP,IGRD
D1MENSION JPSL(8)
COMMON /MDATA/ MREC,MLOC,M5(512)
BYTE MF
INTEGER MREC
COMMON /MDATA/ MREC,MLOC,M5(512)
BYTE MF
INTEGER MREC

COMMON /BFCNTL/ KM,KN,MAPSEL(5),MLVL(4)
COMMON /CNTKST/ KCH(80),KINDX
COMMON /BLKDAT/ LBP(5),LBP(4,4)
COMMON /GRYSC/ KRP(256),IRAP(256),IDMP(4096)
COMMON /RESET/ MK(4),MORM,MBIAS
COMMON /ZIGZAG/ IZZ(1024)
COMMON /LVLTBL/ LVLT(366)
COMMON /MAPS/ NSQ,NLVL,ICODE(1024),KNRTST(1024),INTENS(1024),
       LEVEL(1024),PATTERN(1024)
BYTE LEVEL,PATTH
COMMON /MSTATS/ KOUNT(2,6),DIFF(2,6),DIFFSO,SUMSO
INTEGER KOUNT
REAL DIFF,DIFFSO,SUMSO
DATA NPSL/80/,MPSL/28,52,12,36,60,20,44,4/
DATA JPSL/4,1,6,3,0,5,2,7/
DATA IDMP/1,1,2,2,1,2,1,1,2,1,2,2,2,1,1/
IFILE=1
MIXBP=0
MM=1
DO 110 J=1,6
   MIXBP=MIXBP,OK.(M*(LBP(J),AND,"1"))
110 MM=2*M
KSF=KSF
KSFT=KSF*KSF
NPSL=(NP-1)/KSF+1

III-61
MAPS Compression: TransMAPS 2-15

IF(IGRD.EQ.0) GO TO 130
DO 120 J=1,8
120  NPSL(J)=(NPSI+JPSL(J))/8
130  KSFH=KSF/2
      FL=16./NL
      FP=16./NP
      JGRD=IGRD
      MREC=1
      WLOC=0
      DO 140 J=1,512
140     NSF(J)=0
      MM=4
      IF(KSF.NE.0) MM=3
      KM=0
      I=0
      DO 170 L=1,4
           DO 160 K=1,5
                DO 150 J=1,4
                     IF(KSF.GT*B) CALL LVLSET
                     USQ=KSF*KSF
                     IF(KSF.EQ.16) NLVL=5
                     IF(KSF.EQ.32) NLVL=6
                     DO 190 J=1,1024
                          ICODE(J)=0
                          KNTRST(J)=0
                          INTENS(J)=0
                          LEVEL(J)=0
                          PATTRN(J)=0
                     190   CONTINUE
                     DO 210 J=1,6
                          DO 200 I=1,2
                               KOUNT(I,J)=0
                       200     CONTINUE
                          DIFF(I,J)=0.DO
210     CONTINUE
                      DIFSQ=0.DO
                      SUNSQ=0.DO
                      RETURN
                      END

III-62
SUBROUTINE RMAPS

Purpose: Establish functional gray-scale manipulation controls

- Establish code (8-bit gray-scale) to contrast space remap
- Establish code to intensity space remap and demap
- Establish intensity reset masks, normalization factor, and rounding bias

CALLED from: SETUPM

COMMON /COMTH/ MAC(16,16),KCM(4,4),TB(4),SF(4),SB(4),
+ KCM(4,5,4),LB(6),KBP(2,9),IBP(2,9),IRSET
COMMON /GRYSC/ KRM(256),IRP(256),IDP(4096)
COMMON /RESET/ MA(4),NORM,NBIAS

DO 130 JSEG=2,9
LBP1=KBP(1,JSEG-1)+1
LBP2=KBP(1,JSEG)+1
DL=LBP2-LBP1
NBP1=KBP(2,JSEG-1)+1
NBP2=KBP(2,JSEG)+1
DN=NBP2-NBP1
IF(LBP1.EQ.LBP2) GO TO 120
F=DN/DL

DO 110 J=LBP1,LBP2
110 IRMP(J)=NBP1+F*(J-LBP1)-0.5
120 IF(LBP2.GE.256) GO TO 200
130 CONTINUE

DO 250 JSEG=2,9
LBP1=IBP(1,JSEG-1)+1
LBP2=IBP(1,JSEG)+1
DL=LBP2-LBP1
NBP1=IBP(2,JSEG-1)+1
NBP2=IBP(2,JSEG)+1
DN=NBP2-NBP1
IF(LBP1.EQ.LBP2) GO TO 220
F=DN/DL

DO 210 J=LBP1,LBP2
210 IRMP(J)=NBP1+F*(J-LBP1)-0.5
220 IF(NBP1.EQ.NBP2) GO TO 240
230 CONTINUE

DO 230 J=NBP1,NBP2
230 IDP(J)=LBP1+F*(J-NBP1)-0.5
240 IF(LBP2.GE.256) GO TO 300
250 CONTINUE
MAPS Compression: TransMAPS 2-17

300 DO 310 J=1,4
310 MSK(J)=0
IR=IRSET
IF((IR.EQ.1HM).OR.(IR.EQ.1HL)) MSK(1)=7777
IF((IR.EQ.1HM).OR.(IR.EQ.1HP).OR.(IR.EQ.1HS)) MSK(2)=7777
IF((IR.EQ.1HM).OR.(IR.EQ.1HP).OR.(IR.EQ.1HT)) MSK(3)=7777
IF((IR.EQ.1HM).OR.(IR.EQ.1HN)) MSK(4)=7777
NORM=1
IF(IR.EQ.1HP) NORM=2
IF(IR.EQ.1HM) NORM=4
NBIAS=NORM/2
RETURN
END
SUBROUTINE ZIGZAG

C Purpose: Establish subframe raster to zigzag conversion table

C CALLED from: SETUPM

COMMON /SFCNTL/ KSFT,KSQT,NPST,NPSL(8),KSFH,FL,FP,JGRD
COMMON /ZIGZAG/ IZZ(1024)
DATA IZZ/1024$0/
DO 120 JZZ=1,KSQT
MZZ=JZZ-1
KMSK=MZZ.AND."525
LMSK=(MZZ/2).AND."525
K=0
L=0
M=1
DO 110 J=1,5
KT=KMSK.AND."1
K=K.OR.(M*KT)
LT=LMSK.AND."1
L=L.OR.(M*LT)
KMSK=KMSK/4
LMSK=LMSK/4
110 M=2*M
120 IZZ(IRST)=JZZ
RETURN
END
SUBROUTINE LVLSET

C Purpose: Establish level-triplet to byte-packed conversion table

Levels (resolution codes) are concatenated as a
three-digit octal number to form the look-up address,
L3L2L1. Maximum value is octal 555 for 32x32 case

C CALLED from: SETUP

C******************************************************************************
C
COMMON /LVLTBL/ LVLT(366)
DIMENSION H(3,6)
DATA LVLT/366*1/
DATA H/1,16,64,1,64,0,1,64,64,1,64,64,0,1,64,64,0,1/
L1=2
DO 140 I1=1,6
L1=I1-1
  DO 130 I2=1,I1
    DO 120 I3=1,I2
      L3=I3-1
      DO 110 IP=1,6
        LV=L1*M(1,1P)+L2*M(2,1P)+L3*M(3,1P)+1
        IF(LVLT(LV).NE.-1) GO TO 110
        L1=LI+1
        LVLT(LV)=LI
      CONTINUE
    CONTINUE
  CONTINUE
120 CONTINUE
130 CONTINUE
140 CONTINUE
RETURN
END
SUBROUTINE SFNACC(LOOPSF)

Purpose: Convert the subframe index, LOOPSF, to the corresponding macro-fidelity control
Determine the center line and pixel of the subframe
Scale to the macro-fidelity coordinate indices
Generate the contrast control matrix base address for the corresponding fidelity control group (1-4)

Called from: MAPS

INTEGER*4 LOOPSF, ISTAR4
COMMON /CONTRL/ MAC(16,16), KCS(4), CS(4), TP(4), SF(4), SB(4),
+ KCM(4,5,4), LBP(6), KBP(2,9), IBP(2,9), IKSET
COMMON /SFCHTL/ KSFT, KSOT, NPSL, MSL, NPSL(8), MPSL(8), KSFH, FL, FP, JGRD
COMMON /CMTRST/ KCM(80), KINDEX
ISF100=LOOPSF-1/NPST ! I#4
NROW=ISTAR4
NPSL=NPST*ISTAR4 ! I#4
IF(JGRD .NE. 0) GO TO 110

Square Grid Partition

K=KSFT*(LEFT-1)+KSFH
L=KSFT*NROW+KSFH
GO TO 150

Staggered Grid Partition

LEFT=LEFT
DO 120 J=1,8
LMP=LFT-NPSL(J)
IF(LMP.LT.0) GO TO 130
LFT=LMP
120
130 K=KSOT*(LEFT-1)+MPSL(J)
L=KSFT*(NROW-1)+KSFH+J
150 MK=FP*K+1
IF(MK.LT.1) MK=1
IF(MK.GT.16) MK=16
ML=FL*L+1
IF(ML.LT.1) ML=1
IF(ML.GT.16) ML=16
KINDEX=20*MAC(MK,ML)=19
RETURN
END
SUBROUTINE SFIN

Purpose: Input and convert next source image subframe
Re-order from subframe raster to zigzag position
Remap gray scales from code to contrast and intensity spaces
Initialize level and pattern assignments

Called from: MAPS

COMMON /SFDATA/ ISF(1024)
BYTE ISF
COMMON /GRYSCIL KRNP(256), IRNP(256), IDMP(4096)
COMMON /ZIGZAG/ IZZ(1024)
COMMON /MAPSSF/ NSQ, NLVL, ICODE(1024), KNTRST(1024), INTENS(1024),
+ LEVEL(1024), PATTERN(1024)
BYTE LEVEL, PATTERN
READ (2) (ISF(J), J=1, NSQ)
DO 110 J=1, NSQ
N=ISF(J)
IF(N.LT.0) N=N+256
JZZ=IZZ(J)
ICODE(JZZ)=N
N=N+1
K=KRNP(N)
KNTRST(JZZ)=K
I=IRNP(N)
INTENS(JZZ)=I
LEVEL(JZZ)=0
PATTERN(JZZ)=0
110 CONTINUE
RETURN
END
SUBROUTINE SFMAPS

C Purpose: MAPS compression of source image subframe
Loop on levels
Loop on MAPSel quads (in zigzag order)
Form contrasts and sign-sort vector
Test contrasts and form MAPSel (if required)
Recursion on completed MAPSels
Accumulate optimum bias and performance statistics
Transfer MAPSels to output buffer
Output 512-byte MAPSel records asynchronously

CALLED from: MAPS

CALLS: ODIFF, THRESH, MAPOUT

COMMON /MDDATA/ MREC,MLOC,MSF(512)
BYTE MSF
INTEGER*4 MREC
COMMON /NTEMP/ NN,KM,MAPSEL(5),MLVL(4)
COMMON /CNTRST/ KCMPS(80),KINDX
COMMON /BLKPAT/ LBPT(5),LBBP(4,4)
COMMON /GRISCL/ KNMP(256),IRMP(256),IDMP(4096)
COMMON /MAPSSF/ MS,MLVL,ICODE(1024),KNTRST(1024),INTENS(1024),
+ LEVEL(1024),PATTEN(1024)
BYTE LEVEL,PATTM
COMMON /MSTATS/ KOUNT(2,6),DIFF(2,6),DIFFSQ, SUMSQ
INTEGER*4 KOUNT
REAL*4 DIFF,DIFFSQ,SUMSQ
COMMON /QUAD/ KT(4),IT(4),NT(6),LQ,LVLP,NN
DIMENSION NSTEP(6)
INTEGER*4 LSTPM
DATA NSTEP/i,4,16,64,256,1024/
MTRNS=MLVL=1
DO 130 LTRNS=1,MTRNS
LSTP=MSTP(LTRNS+1)
MSTP=MSTP(LTRNS)
LVLP=LTRNS
DO 120 LQ=1,LSTP
LO=1, LQP=LQAD
IMAPS=LQAD
DO 110 MPSSEL=1,4
IF(LEVEL(IMAPS).NE.LVL) GO TO 120
KT(MPSSEL)=KNTRST(IMAPS)
IT(MPSSEL)=INTENS(IMAPS)
IMAPS=IMAPS+MSTP
110 CALL ODIFF
CALL THRESH
120 CONTINUE
130 KINDX=KINDX+4
MAPS Compression: TransMAPS 2-23

200
LQ=1
LVL=LEVEL(LQ)
LVL= LVL+1
INT=INTENS(LQ)+1
KODE=IDMP(INT)
IF(LVL.GT.0) GO TO 210
IC=ICODE(LQ)
ISTAR4=IC
SUMSQ=SUMSQ+ISTAR4*ISTAR4
1 I*4
IF(KONST.LT.0) GO TO 210
KODE=KODE.AND.16
LQ=LQ+1
KBPT=0
GO TO 300

210
KBT=KSTEP(LVL)
IPAT=PATTYN(LQ)
JPAT=1
KBPT=LBPT(LVL)
IF(KBPT.NE.0) KODE=KODE.AND.1374
DO 230 JQUAD=1,4
IF(KBPT.NE.0) JPAT=IQBP(JQUAD,IPAT+1)
IC=ICODE(LQ)
ISTAR4=IC
SUMSQ=SUMSQ+ISTAR4*ISTAR4
1 I*4
DIFFSQ=DIFFSQ+ISTAR4*ISTAR4
1 I*4
KOUNT(JPAT,LVL)=KOUNT(JPAT,LVL)+1
LQ=LQ+1
220 CONTINUE
230 CONTINUE
300 KM=KM+1
IF(KBPT.NE.0) KODE=KODE.OR.IPAT
MAPSEL(KM+1)=KODE
MLVL(KM)=LVL
IF(KM.NE.KM) GO TO 310
CALL MAPSUT
310 IF(LQ.LE.NSQ) GO TO 200
RETURN
END
SUBROUTINE QDIFF

Purpose: Generate quad contrasts and sign-sort vector

Form all six quad contrasts: 3-2 3-1 2-0 1-0 2-1 3-0
Pack signs of contrasts in left-right order shown
for sign-sort vector
Lowest two bits automatically give pattern code

CALled from: SFMAPS

COMMON /QUAD/ KT(4),IT(4),NT(6),LQ,LQLP,NN
MT(1)=KT(4)=KT(3)
MT(2)=KT(4)=KT(2)
MT(3)=KT(3)=KT(1)
MT(4)=KT(2)-KT(1)
MT(5)=KT(3)=KT(2)
MT(6)=KT(4)=KT(1)
NN=0
DO 110 J=1,6
NN=NN+1
IF(MT(J).LT.0) NN=NN+1
CONTINUE
RETURN
END
SUBROUTINE THRESH
C Purpose: Test contrasts and form new composite MAPS (if required)
C Based on sign-sort vector index NSRT:
   NSRT gives contrast indices in following order:
      (Extreme Middle step Lower step Upper step)
   ISRT gives intensity indices in increasing-value order
C Demap new MAPS from intensity space; then remap to
   contrast space
C CALLED from: SFMAPS
C---------------------------------------------------------------
COMMON /CNTRST/ KCMT(80),KINDX
COMMON /GRISCL/ KRP(256),IRMP(256),IDMP(4096)
COMMON /RESLT/ MSK(4),NORM,NBIAS
COMMON /MAPSSF/ NSQ,NLVL,ICODE(1024),KNTRST(1024),INTENS(1024), + LEVEL(1024),PATTERN(1024)
BYTE LEVEL,PATTERN
COMMON /QUAD/ K4(4),MT(6),LQ,LVLP,NN
DIMENSION NSRT(4,64),ISRT(4,64)
DATA NSRT/6,5,4,1, 4*0, 6,5,3,2, 4*0, 2,3,4,1, 20*0, 1,4,3,2, + 4*0, 2,3,5,6, 4,1,5,6, 1,4,5,6, 3,2,5,6, 8*0, 4,1,3,2, 28*0, + 5,6,3,2, 5,6,1,4, 12*0, 3,2,1,4, 3,2,4,1, 12*0, 5,6,4,1, + 5,6,2,3, 20*0, 4,1,2,3, 8*0, 3,2,6,5, 1,4,6,5, 4,1,6,5, + 2,3,6,5, 4*0, 1,4,2,3, 20*0, 6,5,2,3, 4*0, 2,3,1,4, 4*0, + 6,5,1,4/
DATA ISRT/1,2,3,4, 4*0, 1,3,2,4, 4*0, 2,1,3,4, 20*0, 3,1,2,4, + 4*0, 2,3,1,4, 2,3,4,1, 3,2,1,4, 3,2,4,1, 8*0, 1,3,4,2, 28*0, + 3,1,4,2, 3,4,1,2, 12*0, 3,4,2,1, 1,2,4,3, 12*0, 2,1,4,3, + 2,4,1,3, 20*0, 2,4,3,1, 8*0, 1,4,2,3, 4,1,2,3, 1,4,3,2, + 4,1,3,2, 4*0, 4,2,1,3, 20*0, 4,2,3,1, 4*0, 4,3,1,2, 4*0, + 4,3,2,1/
MNP=MNP+1
KX=KINDX-1
DO 120 J=1,4
   MDP=NSRT(J,MNP)
   KX=KX+1
   IF (IABS(MT(NDX)),GT,KCMT(KX)) GO TO 130
110 CONTINUE
   NEW=MBIAS
   DO 120 J=1,4
      IDX=ISRT(J,MNP)
      NEW=NEW+MSK(J).AND.INT(IDX)
      KNEW=IDMP(NEW+1)
      KNTRST(LQ)=KRP(NEW)
      INTENS(LQ)=NEW
      LEVEL(LQ)=LVL(PATTERN(LQ))=NN.AND.3
120 RETURN
END
SUBROUTINE NAPOUT

Purpose: Transfer completed MAPSel multiplets to output buffer
Pack level multiplets to byte (4 if 8x8, 3 if 16x16 32x32)
Transfer intensity/pattern codes to following bytes (4 3)
Output 512-byte MAPSel stream records asynchronously

CALLED from: SFNAPS

COMMON /MDATA/ NREC,MLOC,MSF(512)
BYTE MSF
INTEGER#4 MREC
COMMON /MTemp/ MM,KM,MAPSEL(5),MLVL(4)
COMMON /LVLTBL/ LVLT(366)
IF(MM.EQ.3) GO TO 120
M=1
LV=0
DO 110 J=1,KM
LV=LV+M#MLVL(J)
M=M#4
MAPSEL(1)=LV
GO TO 140
120 M=1
LV=1
DO 130 J=1,KM
LV=LV+M#MLVL(J)
M=M#4
MAPSEL(1)=LVLT(LV)
130 KM=KM+1
DO 150 J=1,KM
MT=MAPSEL(J)
IF(MT.GT.127) MT=MT-256
MLOC=MLOC+1
MSF(MLOC)=MT
IF(MLOC.LT.512) GO TO 150
NREC=NREC+1
WRITE (3'MREC) MSF
NLOC=0
150 CONTINUE
KM=0
RETURN
END
SUBROUTINE LSTREC

C Purpose: Transfer remaining MAPSeis to output buffer and complete MAPSel stream

C CALLED from: MAPS

COMMON /MDATA/ NREC,MLOC,MSF(512)
BYTE MSF
INTEGER*4 NREC
COMMON /MTEMP/ NM,KM,MAPSEL(5),MLVL(4)
COMMON /LVLTBL/ LVLT(366)
IF(NM.LE.0) GO TO 200
IF(NM.EQ.3) GO TO 120
N=1
LV=0
DO 110 J=1,KM
LV=LV+N*MLVL(J)
110 N=4*M
MAPSEL(1)=LV
GO TO 140
120 N=1
LV=1
DO 130 J=1,KM
LV=LV+N*MLVL(J)
130 N=8*M
MAPSEL(1)=LVLT(LV)
140 KM=KM+1
DO 150 J=1,KM
MT=MAPSEL(J)
IF(MT.GT.127) MT=MT-256
MLOC=MLOC+1
MSF(MLOC)*=MT
IF(MLOC.LE.512) GO TO 150
NREC=NREC+1
WRITE (3*MREC) MSF
MLOC=0
CONTINUE
150 IF(MLOC.LE.0) GO TO 300
NLP=MLOC+1
DO 210 J=NLP,512
210 MSF(J)=0
NREC=NREC+1
WRITE (3*MREC) MSF
RETURN

END
SUBROUTINE SUMMARY

* Purpose: Determine optimum pattern biases where designated and report overall compression and fidelity performance.

CALLING from: MAPS

COMMON /HEADER/ IFILE, INAME(*), NL, NP, MB, KSF, IGRO, NS, MC, MIXBP,
+    IBV(5, 2), IPAD(7)
    BYTE INAME
    INTEGER*4 NS, MC
    COMMON /DATA/ NREQ, MLOC, NSF(512)
    BYTE NSF
    INTEGER*4 NREC
    COMMON /MSTATS/ KOUNT(2, 6), DIFF(2, 6), DIFFSQ, SUMSQ
    INTEGER*4 KOUNT
    REAL*8 DIFF, DIFFSQ, SUMSQ
    DIMENSION NSIZE(5)
    INTEGER*4 LB1T, ISTAR4
    REAL*8 TEMP0, PIX, BITI, BTM
    DATA NSIZE/4, 16, 64, 256, 1024/
    MC=KOUNT(1, 1)+KOUNT(2, 1)
    KOUNT(1, 1)=MC
    DO 120 J=1, 5
        JJ=J+1
        DO 110 I=1, 2
            IOPT=0
            IF(KOUNT(I, JJ).EQ.0) GO TO 110
            TEMP0=KOUNT(I, JJ)
            OPT=DIFF(I, JJ)/TEMP0
            IOPT=OPT+0.5
            IF(OPT.LT.0.) IOPT=OPT-0.5
            DIFFSQ=DIFFSQ-2.*IOPT*DIFF(I, JJ)+IOPT*OPT*TEMP0
            110 1BY(J, 1)=IOPT
            KOUNT(1, JJ)=(KOUNT(1, JJ)+KOUNT(2, JJ))/NSIZE(J)
            120 CONTINUE
    TYPE 500, NREC=2, MLOC
    500 FORMAT(/1X, "MAPS FILE CONTAINS", 16, " 512-BYTE RECORDS PLUS",
+        I4, " BYTES IN THE LAST")
    NLVL=4
    IF(KSF.EQ.16) NLVL=5
    IF(KSF.EQ.32) NLVL=6
    TYPE 510, (L=1, L=1, NLVL)
    510 FORMAT(/1X, "MAPS FILE DISTRIBUTION", 1X, 3X, "LEVEL:", 17, 519)
    TYPE 520, (KOUNT(1, L), L=1, NLVL)
    520 FORMAT(3X, "COUNT", 619)
    NLVL=NLVL-1
    TYPE 530, (IBV(J, 1), J=1, NLVL)
    530 FORMAT(/3X, "OPTIMAL BIAS: ", 17, 419)
    TYPE 540, (IBV(J, 2), J=1, NLVL)
    540 FORMAT(17X, "", 17, 419)
MAPS Compression: TransMAPS 2-29

ISTAR4=NL
PIX=NP*ISTAR4
BITI=NB*PIX
LBIT=(MC+2)/3
IF(KSF.EQ.0) LBIT=(MC+3)/4
BITM=8*(MC+LBIT)
CR=BITI/BITM
BPP=BITM/PIX
ERROR=100.*(DIFFSQ/SUMSQ)

TYPE 550,CR,BPP,ERROR

550 FORMAT(/,1X,"COMPRESSION RATIO:"F8.3," : 1","/,,1X,
+ "BITS/PIXEL:"F6.5," /,,1X,"MEAN SQUARE ERROR:"F9.5." %")
RETURN
END
SECTION NINE

TRANSMAPS MODULE #3: MAPS DECOMPRESSION AND RESOLUTION IMAGE FORMATION

9.1 Program Characteristics

Program Names: DMAPS or DM

Subroutines: USERD
             FILES
             SETUPD
             GAZGIZ
             LDCSET
             SFDMAP

Files: MAPS.DAT (input)
       DMAPS.DAT (output)
       LEVEL.DAT (output)

Task Build Options: MAXBUF = 1024

Task Size: 18720

9.2 Source Listing

The COMMENT-annotated source listing for DMAPS follows:
MAPS Decompression and Level Image Formation: TransMAPS 3-1  DMAPS

PROGRAM DMAPS

<table>
<thead>
<tr>
<th>TransMAPS Module #3: MAPS Decompression &amp; Level Image Formation</th>
</tr>
</thead>
<tbody>
<tr>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>Control Data Corporation - 1982</td>
</tr>
<tr>
<td>----------------------------------------------------------------</td>
</tr>
</tbody>
</table>

Files: Unit Name Content From/To Type
------- -------- ------------------------ ---- --- -------
In  2 MAPS MAPS stream, 512 byte rec. MAPS FIXED SEQ.     
Out 3 DMAPS MAPS Decompressed image, ADAPT FIXED SEQ. or DIFFER or RASTER     
Out 4 LEVEL Level (Resolution) image, ADAPT FIXED SEQ. or RASTER

User Interaction: No parameter inputs required

Program Structure:

PROGRAM DNAPS
  CALL USEKD Dummy routine for future user interaction
  CALL FILESD Open files and read/write headers
  CALL SETUPD Establish control tables
    CALL GAZGIZ Establish zigzag to raster conversion table
    CALL LOCSET Establish byte to level-multiplet table
    Establish intensity/pattern decode table
  Loop on subframes
    CALL SFDMAP Convert MAPS stream to decompressed end
    resolution (level) images by subframe

COMMON Block Communication:

/BLKMAT/ Block/pattern mode by level Length: 21
  SETUPD, SFDMAP
  LBPT(5) [1*2] Block(0)/pattern(1) for levels 1-5
  LBPT(4,4) [1*2] Low/High index by Quadrant,Pattern

/DMAPSF/ Decompression and level image output Length: 1025
  FILESD, SFDMAP
  NSQ [1*2] Number of total pixels/subframe
  ISF(1024) [Byte] Decompressed subframe array
  LSF(1024) [Byte] Level image subframe array

/GAZGIZ/ Zigzag to raster conversion Length: 1024
  GAZGIZ, SFDMAP
  NZZ(1024) [1*2] Zigzag to raster conversion table

III-78
MAPS Decompression and Level Image Formation: TransMAPS 3-2 DNAPS

/HEADER/ Standard MAPS file header

**Length:** 32

- **FILE**
  - [I*2] File type
- **INAME(8)**
  - [byte] User-selected image name
- **ML**
  - [I*2] Number of lines in source image
- **MP**
  - [I*2] Number of pixels in source image
- **MB**
  - [I*2] Number of bits/pixel in source image
- **KSF**
  - [I*2] Kind of subframe 8x8, 16x16, 32x32
- **IGRD**
  - [I*4] Total subframe count
- **MC**
  - [I*4] MAPSEL count
- **MIXBP**
  - [I*2] Packed block(0)/pattern(1) mode (rt-lft)
- **IBV(5,2)**
  - [I*2] Optimal pattern biases by level, low/high
- **IPAD(7)**
  - [I*2] Space for future extension

/LDCODE/ Byte-packed Level decode

**Length:** 1024

- **LDC(4,256)**
  - [I*2] Byte to level-multiplet decoding table

/MDATA/ MAPSel stream data

**Length:** 260

- **MLOC**
  - [I*2] Current location in MAPS input buffer
- **MSF(512)**
  - [byte] MAPSel stream input buffer
- **MLVL**
  - [I*2] Current packed level byte
- **MN**
  - [I*2] Number of levels/byte 8x8(4), 16x16 or 32x32(3)
- **KM**
  - [I*2] Current position of level in byte (1-MN)

/MDCODE/ Intensity/pattern decode

**Length:** 2560

- **MDC(256,4,5)**
  - [I*2] Intensity/pattern decoding table by intensity/pattern byte, quadrant, level

---

**COMMON /HEADER/**

- **FILE, INAME(8), ML, MP, MB, KSF, IGRD, NS, MC, MIXBP,**
- **IBV(5,2), IPAD(7)**
- **BYTE INAME**
- **INTEGER*4 NS, MC**
- **INTEGER*4 LOOPSF**
- **CALL USERD**
- **CALL FILESD**
- **TYPE 500, INAME, ML, MP**
- **FORMAT(/,1X,"MAPS DECOMPRESSING IMAGE ",8A1,"","IS," LINES BY",**
- **IS, "PICTURES")**
- **CALL SETUP**
- **DD 110 LOOPSF=1, NS**
- **CALL SFDMAP**
- **CLOSE(UNIT=2)**
- **CLOSE(UNIT=3)**
- **CLOSE(UNIT=4)**
- **END**
SUBROUTINE USERD

C
C Purpose: User interaction for MAPS decompression and level image formation
C Dummy routine for possible future extensions
C
C CALLed from: DMAPS
C
C------------------------------------------------------------------------

500 FORMAT(/,1X,46(1H*),/,1X,"*",44X,"*",/1X,
+ "* MAPS DECOMPRESSION/RESOLUTION IMAGE MODULE ",/1X,"*",44X,
+ "*",/1X,46(1H*),/1X,"NO USER INPUTS REQUIRED")
RETURN
END
SUBROUTINE FILESD

Purpose: Open files, read/write standard MAPS headers

CALLED from: DMAPS

---

COMMON /HEADER/ IFILE,INAME(6),NL,MP,NB,KSF,NS,MC,MIXBP,
+   IBV(5,2),IPAD(7)
BYTE INAME
INTEGER*4 NS,MC
DIMENSION JHEAD(32)
EQUIVALENCE (JHEAD(1),IFILE)
COMMON /MDATA/ MLOC,NSF(512),MLVL,MM,KK
BYTE NSF
COMMON /DMAPS/ NSQ,ISF(1024),LSF(1024)
BYTE ISF,LSF
DIMENSION JSF(512)
EQUIVALENCE (JSF(1),ISF(1))
DATA ISF/1Q24*O/,LSF/1024*O/
+   RECORDTYPE="FIXED"
READ (2) JHEAD
IF(IFILE.EQ.1) GO TO 110
  TYPE 500,IFILE
  500 FORMAT(/,IX,"*** FILE TYPE",13," NOT MAPSel STREAM")
  STOP
110 READ (2) NSF
  MLOC=1
  MLVL=NSF(1)
  IF(MLVL.LT.0) MLVL=MLVL+256
  MLVL=MLVL+1
  MM=4
  IF(KSF.NE.0) MM=3
  NS=KSF*NSF
  LSG=NS/4
  DO 120 J=1,32
  120 JSF(J)=JHEAD(J)
  OPEN(UNIT=3,TIP="NEW",NAME="DMAPS",FORM="UNFORMATTED",
+     RECORDTYPE="FIXED",RECORDSIZE=LSG)
  JSF(1)=2
  WRITE (3) (ISF(J),J=1,NSF)
  OPEN(UNIT=4,TIP="NEW",NAME="LEVEL",FORM="UNFORMATTED",
+     RECORDTYPE="FIXED",RECORDSIZE=LSG)
  JSF(1)=3
  WRITE (4) (ISF(J),J=1,NSF)
  DO 130 J=1,32
  130 JSF(J)=0
RETURN
END
SUBROUTINE SETUPD

C Purpose: Establish decompression control tables
C Establish zigzag to subframe raster conversion table
C Establish byte to level multiple resolution decode table
C Establish intensity/pattern byte decode table
C
C CALLED from: DMAPS
C CALLS: GA2GIZ, LDCSET

C------------------------------------------------------------------------

COMMON /HEADEN/ IFILE,INAME(N),NL,NP,NB,KSF,IGRD,NS,NC,MIXBP,
+ 1BV(5,2),1PAD(7)
BYTE INAME
INTEGER*4 NS,NC
COMMON /MDCODE/ MDC(256,4,5)
BYTE MDC
COMMON /BLKPAT/ LBPT(5), IOBP(4,4)
DATA IOBP/1,1,2,2, 2,1,2,1, 1,2,1,2, 2,2,1,1/
KSF=KSF
CALL GA2GIZ(KSF)
CALL LDCSET(KSF)
M=2
DO 110 J=1,5
LBPT(J)=(MIXBP/M).AND.*1
110  M=2*M
DO 150 LV=1,5
KBPT=LBPT(LV)
  DO 140 JW=1,4
     DO 130 I=1,256
        M=I-1
        IF(KBPT.EQ.0) GO TO 120
        IPAT=M.AND.*3
        INT=M-IPAT
        IBP=Iobp(JW,IPAT+1)
        JBv=1BV(LV,IBP)
        M=INT+JBV
        IF(M.LT.0) M=0
        IF(M.GT.255) M=255
120  IF(M.GT.255) M=255
130  MDC(I,JW,LV)=M
140  CONTINUE
150  CONTINUE
RETURN
END

III-82
SUBROUTINE GAZGIZ(KSFT)

Purpose: Establish zigzag to subframe raster conversion table

CALLed from: SETUP

COMMON /GAZGIZ/ NZZ(1024)
DATA NZZ/1024*0/
KSQT=KSFT*KSFT
DO 120 JZZ=1,KSQT
MZZ=JZZ-1
KMSK=MZZ,AND."525
LMSK=(MZZ/2),AND."525
K=0
L=0
M=1
DO 110 J=1,5
KT=KMSK,AND."1
KOL=OK.(N*KT)
LT=LMSK,AND."1
L=LOL.(N*LT)
KMSK=KMSK/4
LMSK=LMSK/4
110 IN=2*M
120 NZZ(JZZ)=IN
RETURN
END

III-83
SUBROUTINE LDCSET(KSFT)

C Purpose: Establish byte to level multiplet resolution decode table
by MAPSel sequence (1-4 for 8x8, 1-3 for 16x16 & 32x32) and byte value

C CALLED from: SETUPUD

COMMON /LDCO.L/
LDC(4,256)
DATA LDC/10244-1/
DATA LVLT/366*-I/
DATA M/1,6,64, 1,64, 8,1,64, 8,64,1, 8,64, 8,1/
IF(KSFT,NE.8) GO TO 200
DO 120 I=1,256
LPACK=I-1
MT=1
110 DO 110 J=1,4
LDC(J,1)=(LPACK/MT).AND.3
MT=4*MT
120 CONTINUE
GO TO 300
200 LI=2
DO 240 LI=1,6
L2=LI-1
DU 220 LI=1,12
L3=LI-1
DO 210 IP=1,6
LV=LI*M(1,IP)+L2*M(2,IP)+L3*M(3,IP)+1
IF(LVLT(LV),NE.-1) GO TO 210
LI=LI+1
LVLT(LV)=LI
CONTINUE
210 CONTINUE
220 CONTINUE
230 CONTINUE
240 CONTINUE
DO 260 IT=1,366
L=LVLT(IT)
IF(L.EQ.-1) GO TO 260
LPACK=IT-1
MT=1
125 DO 250 J=1,3
LDC(J,1+1)=(LPACK/MT).AND.7
MT=8*MT
250 CONTINUE
300 CONTINUE
RETURN
END
SUBROUTINE SFDMAP

C Purpose: Convert MAPSel stream to decompressed and resolution (level) images by subframe
  Recursion on MAPSel's (automatic zigzag order)
  Retrieve level code
  Retrieve intensity/pattern byte
  Asynchronously input next MAPSel record as needed
  Loop on MAPSel quadrants
  Convert intensity/pattern byte
  Loop on pixels within quadrant in zigzag order
  Determine subframe raster address
  Set decompressed and level image pixel
  Return for next MAPSel until subframe completed
  Output decompressed and level image subframes

CALLed from: DNAPS

COMMON /NDATA/ MLOC,NSF(512),MLVL,KM,KN
BYTE NSF
COMMON /DNAPSF/ NSO,ISF(1024),LSF(1024)
BYTE ISF,LSF
COMMON /GAZGIZ/ NZZ(1024)
COMMON /LDCODE/ LDC(4,256)
COMMON /MDCODE/ MDC(256,4,5)
BYTE MDC
COMMON /BLKPAT/ LBPT(5),IQBT(4,4)
DIMENSION NSTEP(5)
DATA NSTEP/i,4,16,64,256/
JZZ=0

100 KM=KM+1
   IF(KM.LE.MM) GO TO 120
   KM=1
   MLOC=MLOC+1
   IF(MLOC.LE.512) GO TO 110
   MLOC=1
   READ (2) NSF
   MLVL=NSF(MLOC)
   IF(MVNL.LE.0) MLVL=MLVL+256
   LV=LOC(KM,MLVL)
   MLOC=MLOC-1
   IF(MLOC.LE.512) GO TO 130
   MLOC=1
   READ (2) NSF
   MZ=NSF(MLOC)
   IF(MZ.LE.0) GO TO 200
   JZ=JZZ+1
   ISF(INST)=M1
   LSF(INST)=0
   IF(JZ.GE.NSO) GO TO 300
   GO TO 100

III-85
200  
MSTP=MSTEP(LV)
IPAT=0
IF(LBPT(LV).NE.0) IPAT=M1, AND.*3
L=32*LV+4*IPAT
IF(L.GT.127) L=L-256
IF(M1.LT.0) M1=M1+256
M1=M1+1
DO 220 JQ=1,4
I=MDC(M1,JQ,LV)
   DO 210 J=1,MSTP
      JZZ=JZZ+1
      I$F(I$ST)=1
   CONTINUE
   LSF(I$ST)=L
220   CONTINUE
   IF(JZZ.LT.MSU) GO TO 100
300   WRITE (3) (ISF(J),J=1,M50)
   WRITE (4) (LSF(J),J=1,M50)
RETURN
END
SECTION TEN

TRANS MAPS MODULE #4: MAPS ADAPTIVE IMAGE SMOOTHING

10.1 Program Characteristics

Program Names: ADAPT or AD

Subroutines:
USERA
FILESAD
SETUPA
LINPIX
WGTSET
UNIFRM
GAUSS
SFLOAD
SFUPDT
SFADPT

Files:
DMAPS.DAT (input)
LEVEL.DAT (input)
ADAPT.DAT (output)

Task Build Options: MAXBUF = 1024

Task Size: 27968

10.2 Source Listing

The COMMENT-annotated source listing for ADAPT follows:
MAPS Adaptive Image Smoothing: TransMAPS 4-1

PROGRAM ADAPT

TransMAPS Module #4: MAPS Adaptive Image Smoothing

Control Data Corporation - 1982

Files: Unit Name Content From/To Type
--
In 2 DMAPS MAPS Decompressed image, subframes DMAPS DIRECT
In 3 LEVEL Level (Resolution) image, subframes DMAPS DIRECT
Out 4 ADAPT Adaptively smoothed image, subframes DIFFER FIXED SEG.

User Interaction: In Subroutine USERT

Convolution weighting specification
Dither amplitude specification

Program Structure:

PROGRAM ADAPT
CALL USERT Specify convolution weighting and dither
CALL FILES A Open files, check integrity, read/write headers
CALL SETUPA Establish control tables, subframe partition
CALL LINPIX Establish zigzag to line, pixel, & raster index conversion tables
CALL WGTSET Establish pre-summed convolution weights
CALL UNIFRM(LVL) Establish uniform window weights
or CALL GAUSS(LVL) Establish Gaussian window weights
Loop on subframes rows in line direction
CALL SFLOAD Initialise first subframe in row and its surround
Loop on subframes within row
CALL SFUPDT Update to next subframe & its surround
CALL SFADPT Adaptively smooth MAPSels in subframe

COMMON Block Communication:

/GAZGIZ/ Zigzag to raster conversion Length: 1024
LINPIX,SFADPT

/ZZZ(1024) [1#2] Zigzag to raster conversion table

III-88
MAPS Adaptive Image Smoothing: TransMAPS 4-2

/HEADD/ Standard MAPS file header from DMAPS Length: 32
ADAPT,FILESA,SETUPA

IFILED [I*2] File type
INAMED(8) [Byte] User-selected image name
MLD [I*2] Number of lines in source image
NPD [I*2] Number of pixels in source image
NBD [I*2] Number of bits/pixel in source image
KSFD [I*2] Kind of subframe 8x8 16x16 32x32
IGROD [I*2] Subframe grid: square(0)/staggered(1)
MSD [I*4] Total subframe count
MCD [I*4] MAPSel count
MIXBPD [I*2] Packed block(0)/pattern(1) mode (rt-lft)
IBVD(5,2) [I*2] Optimal pattern biases by level,low/high
IPADD(7) [I*2] Space for future extension

/HEADL/ Standard MAPS file header from LEVEL Length: 32
FILESA

IFILED [I*2] File type
INAMEL(8) [Byte] User-selected image name
MLL [I*2] Number of lines in source image
NPL [I*2] Number of pixels in source image
NBL [I*2] Number of bits/pixel in source image
KSFL [I*2] Kind of subframe 8x8 16x16 32x32
IGRDl [I*2] Subframe grid: square(0)/staggered(1)
MSL [I*4] Total subframe count
MCL [I*4] MAPSel count
MIXBPL [I*2] Packed block(0)/pattern(1) mode (rt-lft)
IBVL(5,2) [I*2] Optimal pattern biases by level,low/high
IPADL(7) [I*2] Space for future extension

/LPZ/ Zigzag to line/pixel conversion Length: 2048
LINPIX,GTSET,SFADPT

LZZ(1024) [I*2] Zigzag to line conversion table
KZZ(1024) [I*2] Zigzag to pixel conversion table

/SFDATA/ Adaptive image subframe data Length: 512
FILESA, SFADPT

ISFA(1024) [Byte] Adaptive image subframe array

/SFINPT/ Input image (DMAPS,LEVEL) control Length: 12
SETUPA, SFLOAD, SFUPDT

KIST [I*2] First pixel in buffer
KIST [I*2] Last pixel in buffer
LIST [I*2] First line in buffer
LLST [I*2] Last line in buffer
KREC [I*4] Record number of current subframe
KBCK [I*4] Record number of adjacent prior row subframe
KFWD [I*4] Record number of adjacent following row subframe
NREC [I*4] Total number of input file records
MAPS Adaptive Image Smoothing: TransMAPS 4-3

/C/SFLOOP/ Subframe control parameters
Length: 6
ADAPT,SETUPA,LINPIX,SFUPDT,SLADPT

KSF [I#2] Subframe size: edge
NSQ [I#2] Subframe size: number of total pixels
NLS [I#2] Number of subframes in line direction
WPS [I#2] Number of subframes in pixel direction
KLS [I#2] Current subframe index in line direction
KPS [I#2] Current subframe index in pixel direction

/C/SHARE/ DMAPS/LEVEL data; shared with weight Length: 2210
window during weight table generation
WGSET,UNIFORM,GAUSS,
SFLOAD,SFUPDT,SLADPT

ISFD(65,34) [byte] DMAPS subframe and surround data array
ISFL(65,34) [byte] LEVEL subframe and surround data array
Equivalence with:
WNDW(31,31) [R#4] Convolution weight window

/C/WGTDIT/ Convolution and dither selections Length: 5
USERA,WGSET,GAUSS,SLADPT

KCW [I#2] Kind of convolution weight:
SIG [R#4] Multiple of Gaussian sigma at
DIT [R#4] Dither amplitude
window corner

/C/WGTTBL/ Pre-summed convolution weights Length: 6120
WGSET,SLADPT

WGT(9,340) [R#4] Weight table addressed by:
surround index,
compound level/quadrant-zigzag index

MAPSel Surround Conventions:

Pixel direction
|
| Region Quadrant 1 2 3 4 5 6 7 8 9
|
| Associated Surround Index
|
|----------------------------------|
|
| L | 1 | 2 | 3 | 4 |
| 6 | 0 | 1 | 2 | 3 | 5 | 6 | 7 | 9 | 10 | 11 |
|
| H | 5 | 6 | 7 | 8 |
| 7 | 1 | 4 | 3 | 2 | 8 | 7 | 6 | 12 | 11 | 10 |
|
| V | 9 | 10 | 11 | 12 |
| 10 | 2 | 13 | 14 | 15 | 9 | 10 | 11 | 5 | 6 | 7 |
|
| T | 13 | 14 | 15 | 16 |
| 11 | 3 | 16 | 15 | 14 | 12 | 11 | 10 | 8 | 7 | 6 |
COMMON /MEADD/ IFILED,INAMED(N),MLD,NPD,NBD,KBFD,IGRDD,NSD,NCD,
+ MIXBD,IBVD(5,2),IPADD(7)
BYTE INAMED
INTEGER*4 NSD,NCD
COMMON /SFLOOP/ KSF,NSD,NLS,NPS,KLS,KPS
CALL USERA
CALL FILES
CALL SETUPA
CALL SFLOAD
DO 110 IPS=1,NPS
KPS=IPS
CALL SFUPDT
CALL SFADPT
CONTINUE
120 CONTINUE
CLOSE(UNIT=2)
CLOSE(UNIT=3)
CLOSE(UNIT=4)
END
SUBROUTINE USEK

Purpose: User interaction for MAPS adaptive image smoothing

Convolution weighting

Dither amplitude

CALLED from: ADAPT

COMMON /WGDT/L, KCW, SIG, DIT
DATA KCW/1.0/, SIG/2.0/
DATA DIT/4.0/

TYPE 500

500 FORMAT(1X, 3L11), 1X, 1X, 1X
+ 1x, "MAPS ADAPTIVE SMOOTHING MODULE ",
+ 1x, "\"UNIFORM OR GAUSSIAN? (U G) \"", 1X

ACCEPT 1, LIT

1 FORMAT(10A1)

IF(LIT.EQ.1000) KCW=LIT
IF(LIT.EQ.1010) KCW=LIT
IF(KCW.LE.1000) GO TO 110

TYPE 520, SIG

520 FORMAT(5X, "SIGmA MULTIPLE AT WINDOW CORNER?", F5.1, 2X,
+ "(/ = NO CHNG)")

ACCEPT *, SIG

110 TYPE 530, DIT

530 FORMAT(3X, "RANDOM DITHER?:", /, 5X, "AMPLITUDE?", F5.1, 24X,
+ "(/ = NO CHNG)")

ACCEPT *, DIT

540 FORMAT(1X, 3L11), 1X, 1X, 1X

+ 1x, "REVIEW? (Y OR N) \"", 1X

ACCEPT 1, LIT

IF(LIT.EQ.1000) GO TO 100

RETURN

END
SUBROUTINE FILES

C Purpose: Open files, check integrity, read/write headers

C CALLED from: ADAPT

C-------------------------------------------------------------
COMMON /HEADL/ IFILED,INAMEL(8),NLD,NPD,MBD,KSF,D,IGRDL,NSD,MCD,
  + MIXPL,IVD(5,2),IPADD(7)
BYTE INAMEL
INTEGER 4 NSD,MCD
DIMENSION JHEADL(32)
EQUIVALENCE (JHEADL(1),IFILED)
COMMON /HEADL/ IFILED,INAMEL(8),NLD,NPD,MBD,KSF,D,IGRDL,NSL,MCL,
  + MIXPL,IVD(5,2),IPADD(7)
BYTE INAMEL
INTEGER 4 NSL,MCL
DIMENSION JHEADL(32)
EQUIVALENCE (JHEADL(1),INAMEL)
COMMON /SFDATA/ ISFA(1024)
BYTE ISFA
DIMENSION JSF(512)
EQUIVALENCE (JSF(1),ISFA(1))
OPEN (UNIT=2, TYPE='OLD', NAME='DMAPS', FORM='UNFORMATTED',
  + RECORDTYPE='FIXED', ACCESS='DIRECT')
READ (2'1) JHEADL
IF (IGRDL,NE.0) GO TO 210
OPEN (UNIT=3, TYPE='OLD', NAME='LEVEL', FORM='UNFORMATTED',
  + RECORDTYPE='FIXED', ACCESS='DIRECT')
READ (3'1) JHEADL
IF (IGRDL,NE.0) GO TO 210
DO 110 J=1,8
IF (INAMEL(J).NE.INAMEL(J)) GO TO 220
110 CONTINUE
IF (NSD,NE.NSL) GO TO 220
IF (MCD,NE.MCL) GO TO 220
KSQ=KSF*KSDF
LSQ=KSQ/4
OPEN (UNIT=4, TYPE='NEW', NAME='ADAPT', FORM='UNFORMATTED',
  + RECORDTYPE='FIXED', RECORDSIZE=LSQ)
DO 120 J=1,32
120 JSF(J)=JHEADL(J)
JSF(1)=4
WRITE (4) (ISFA(J),J=1,KSQ)
GO TO 300
110 CONTINUE
STOP
210 TYPE 510
FORMAT (/1X,*** NON-SQUARE GRID, ADAPT PRECLUDED*)
STOP
220 TYPE 520,INAMEL,INAMEL,NLD,NPD,MBD,KSF,D,IGRDL,NSD,MCL,
  + MIXPL,IVD(5,2),IPADD(7)
STOP
300 CONTINUE
RETURN
END

III-93
SUBROUTINE SETUPA

Purpose: Establish adaptive smoothing control tables

Establish inter-subframe position controls

Establish zigzag to line, pixel, and subframe raster conversion tables

Establish pre-summed convolution weights

CALLED from: ADAPT

CALLS: LINPIX, WGTSET

COMMON /HEADD/ IFILED, INAMED(8), NLD, NPD, NBD, KSFD, IGRDD, NSD, MCD,
        MIXBD, IBVD(5, 2), IPADD(7)
BYTE INAMED
INTEGER*4 MBD, MCD
COMMON /SFLOOP/ KSF, MSO, MLS, NPS, KLS, KPS
COMMON /SFINPT/ KIST, KLST, LIST, LLST, KREC, KBCK, KFWD, NREC
INTEGER*4 KREC, KBCK, KFWD, NREC
KSF=KSFD
MSO=MSO*KSF
MLS=(NLD-1)/KSF+1
NPS=(NPD-1)/KSF+1
KIST=KSF+2
KLST=2*KSF+1
LIST=2
LLST=KSF+1
KREC=1
KBCK=KREC-NPS
KFWD=KREC+NPS
NREC=NSO+1
CALL LINPIX
CALL WGTSET
RETURN
END

III-94
SUBROUTINE LINPIX

Purpose: Establish zigzag to line, pixel, and subframe raster
index conversion tables

Called from: SETUP

COMMON /SFLOUP/ KSF,NSQ,MLS,MP5,KLS,KPS
COMMON /GAZGIZ/ MZ2(1024)
COMMON /LPZZ/ LZZ(1024),K2Z(1024)
DATA NZZ/1024*0/
KSF=KSF
DO 120 JZZ=1,1024
MZZ=JZZ-1
KMSK=MZ2.AND.525
LMSK=(MZZ/2).AND.525
K=0
L=0
M=1

DO 110 J=1,5
KT=KMSK.AND.1
K=K.OR.(K*KT)
LT=LMSK.AND.1
L=L.OR.(M*LT)
KMSK=KMSK/4
LMSK=LMSK/4
M=2*M
LZZ(JZZ)=L+2
KZZ(JZZ)=K+2
IRST=KSF*L+h+1
110
120 NZZ(JZZ)=IRST
RETURN
END

111-95
SUBROUTINE WGTSET

Purpose: Establish pre-summed convolution weights

Loop on levels of target MAPSel
   Establish convolution weight window:
      uniform or Gaussian
   Loop on pixels in first MAPSel quadrant; zigzag order
   Determine line and pixel ranges for 9 distinct
   MAPSel and surround regions
   Loop on regions
      Loop on lines, pixels within regions
         Sum convolution weights over region
         Set weight by region and pixel zigzag index

CALLED from: SETUPA

CALLS: UNIFRM, GAUSS

COMMON /WGTDIR/ KCW,SIG,DIT
COMMON /SHARE/ ITEMP(2210)
DIMENSION WNDW(31,31)
EQUIVALENCE (#NDW(1,1),ITEMP(1))
COMMON /LPZZ/ LZZ(1024),KZZ(1024)
COMMON /WGTBL/ WGT(9,340)
DIMENSION LW1(3),LWL(3),KW1(3),KWL(3)
DIMENSION MQSZ(5),LWC(5),LWMDW(5)
DATA MQSZ/1,4,16,64,256/,LWC/1,2,4,8,16/,LWMDW/1,3,7,15,31/
I=0
LW1(1)=1
KW1(1)=1
DO 160 LV=2,5
LVL=LVL
IF(KCW.NE.1) CALL UNIFRM(LVL)
IF(KCW.EQ.1) CALL GAUSS(LVL)
MQ=MQSZ(LV)
LWCT=LWC(LV)
LWL(3)=LWMDW(LV)
KWL(3)=LWMDW(LV)
   DO 150 JZZ=1,MQ
      IZ=I+1
      LZZ=JZZ
      K=KZZ(JZZ)-1
      LW1(1)=LWCT-L
      LW1(2)=LW1(1)+1
      LW1(3)=LW1(1)+LWCT
      LW1(4)=LW1(1)+LWCT
      LW1(5)=LW1(1)+LWCT
      LW1(6)=LW1(1)+LWCT
      LW1(7)=LW1(1)+LWCT
      LW1(8)=LW1(1)+LWCT
      LW1(9)=LW1(1)+LWCT
      I=0
   )
DO 140 LREG=1,3
LW1ST=1(LREG)
LWLST=1=L(LREG)
DO 130 KREG=1,3
KW1ST=KW1(KREG)
KWLST=KW1(LREG)
JW=JW+1
WT=0.
IF((LW1ST.GT.LWLST).OR.(KW1ST.GT.KWLST)) GO TO 130
DO 120 LW=LW1ST,LWLST
DO 110 KW=KW1ST,KWLST
WT=WT+WHDW(KW,LW)
CONTINUE
110 CONTINUE
130 CONTINUE
140 CONTINUE
150 CONTINUE
160 CONTINUE
RETURN
END
SUBROUTINE UNIFRM(LVL)

C Purpose: Establish normalized uniform window weights for level LVL
C
C CALLED from: #GTSET

C--------------------------------- f....f..........
COMMON /SHARE/ ITEMP(2210)
DIMENSION WND(31,31)
EQUIVALENCE (WNDW(1,1),ITEMP(1))
DIMENSION LWNDw(5)
DATA LWNDw/1,3,7,15,31/
WNDW=LWNDW(LVL)
WPX=WNW=WNW
WT=1./WPX
DO 120 L=1,NWWD
  DO 110 K=1,NWWD
    WNDW(K,L)=WT
110    CONTINUE
120    CONTINUE
RETURN
END
SUBROUTINE GAUS(LVL)

Purpose: Establish normalized Gaussian window weights for level LVL

CALLED from: 

COMMON /MTIDIT/ KCM.SIG,DIT
COMMON /SHARE/ ITMP(2210)
DIMENSION #NDW(31,31)
EQUIVALENCE (#NDW(1,1),ITMP(1))
DIMENSION LW(5), LWND(5), CORNER(5)
DATA LW/1,2,4,8,16/, LWND/1,3,7,15,31/, CORNER/0,1,3,7,15./
LWCT=LW(LVL)
NWND=LWND(LVL)
C=CORNER(LVL)
CSQ=2.4*C
FSQ2=-SIG*SIG/(2.*CSQ)
SUM=0.
DO 120 L=1,NWND
  Y=LWCT
  DD 110 K=1,NWND
    X=K-LWCT
    XX=X*X
    WT=EXP(FSQ2*(XX+YY))
    SUM=SUM+WT
  110 CONTINUE
  WND(K,L)=T
  120 CONTINUE
DO 220 L=1,NWND
  DD 210 K=1,NWND
    WND(K,L)=WND(K,L)/SUM
  210 CONTINUE
RETURN
END
SUBROUTINE SFLOAD

Purpose: Initialize first target subframe and its surround for each
        new line-direction subframe row
        Blank surround along left image edge
        Load last line of pixels from prior-row subframe
        Load first line of pixels from following-row subframe
        Load target subframe

CALLED from: ADAPT

COMMON /SFLOAD/ K1ST, KLST, L1ST, LLST, KREC, KBCK, KFWD, NREC
INTEGER*4 KREC, KBCK, KFWD, NREC
COMMON /SHAKE/ ISFD(65,34), ISFL(65,34)
BYTE ISFD, ISFL
LP=LLST+1
K=K1ST-1
DO 110 L=1, LP
   ISFD(K, L)=0
110   DO 120 K=1, KLST
         ISFD(K, 1)=0
   120   ISFL(K, L)=0
120   KREC=KREC+1
   140   IF(KBCK.LE.1) GO TO 140
         READ (2'KBCK') (ISFD(K, L), K=K1ST, KLST), L=L1ST, LLST)
         READ (3'KBCK') (ISFL(K, L), K=K1ST, KLST), L=L1ST, LLST)
   130   ISFL(K, 1)=ISFD(K, L1ST)
   150   ISFL(K, LP)=ISFL(K, L1ST)
140   IF(KFWD.GT.NREC) GO TO 160
   160   CONTINUE
         READ (2'KREC') (ISFD(K, L), K=K1ST, KLST), L=L1ST, LLST)
         READ (3'KREC') (ISFL(K, L), K=K1ST, KLST), L=L1ST, LLST)
   110   CONTINUE
SUBROUTINE SFUPDT

Purpose: Update target subframe and its surround

Transfer left surround, prior-line surround, following-
line surround, and target subframe from right surround
area to target area
Load last line of pixels from upper right corner
surround subframe
Load first line of pixels from lower right corner
surround subframe
Load right surround subframe (next target subframe)

Note: "Loads" are replaced by "Blanks" whenever
subframe references would be beyond
image boundaries

CALLED from: ADAPT

Geometry of decompressed and resolution image input buffers:

Pixel Direction ———>

<table>
<thead>
<tr>
<th>record KBC-1</th>
<th>record KBC</th>
<th>record KBC+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>upper left corner</td>
<td>upper surround</td>
<td>upper right corner</td>
</tr>
<tr>
<td>L</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>L</td>
<td>left</td>
<td>right surround</td>
</tr>
<tr>
<td>D</td>
<td>record</td>
<td>record KREC</td>
</tr>
<tr>
<td>Direct</td>
<td>KREC-1</td>
<td>record KREC</td>
</tr>
<tr>
<td>cction</td>
<td>last column</td>
<td>INPUT REGION,</td>
</tr>
<tr>
<td>cction</td>
<td>of pixels</td>
<td>ALL SUBFRAMES</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| lower left corner | lower surround | lower right corner |
| record KFWD-1 | record KFWD | record KFWD+1 |

COMMON /SFLOOP/ KSF,NS0,NS3,NPS,KPS,KPS
COMMON /SFINPT/ KST,LIST,LIST,LST,KREC,KBC,KFWD,NREC
INTEGER*4 KREC,KBC,KFWD,NREC
COMMON /SHARE/ ISFD(65,34),ISFL(65,34)
BYTL ISFD,ISFL
LP=LLST+1
KP=KSF+1

III-101
MAPS Adaptive Image Smoothing: TransMAPS 4-15

DO 120 L=1,LP
KSF=KSF
   DO 110 K=1,KP
      KSF=KSF
      ISFD(K,L)=ISFD(KK,L)
   110
      ISFL(K,L)=ISFL(KK,L)
   120
      CONTINUE
      DO 130 L=1,LP
         ISFD(K1ST,L)=0
      130
      ISFL(K1ST,L)=0
      DO 140 K=1,KLST
         ISFD(K,1)=0
         ISFD(K,LP)=0
      140
      ISFL(K,1)=0
      ISFL(K,LP)=0
      IF(KPS.EQ.WPS) GO TO 200
      KREC=KREC+1
      KBCK=KBCK+1
      KFWD=KFWD+1
      IF(KBCK.LE.1) GO TO 160
      READ (2,KBCK) ((ISFD(K,L),K=K1ST,KLST),L=LIST,LLST)
      READ (3,KBCK) ((ISFL(K,L),K=K1ST,KLST),L=LIST,LLST)
      150
      ISFD(K,1)=ISFD(K,LLST)
      160
      IF(KFWD.GT.KREC) GO TO 160
      READ (2,KFWD) ((ISFD(K,L),K=K1ST,KLST),L=LIST,LLST)
      READ (3,KFWD) ((ISFL(K,L),K=K1ST,KLST),L=LIST,LLST)
      170
      ISFL(K,LP)=ISFL(K,LIST)
      180
      CONTINUE
      READ (2,KREC) ((ISFD(K,L),K=K1ST,KLST),L=LIST,LLST)
      READ (3,KREC) ((ISFL(K,L),K=K1ST,KLST),L=LIST,LLST)
      200
      CONTINUE
      RETURN
      END
SUBROUTINE SFADPT

Purpose: Adaptively smooth all MAPSels > 2x2 in target subframe

Recursion through all MAPSels in target subframe, zigzag order

Establish line and pixel indices for sixteen target and surround elements

Loop on target/surround elements

Retrieve element intensity

Retrieve element level; set activation if surround element no more than one level below target MAPSel

Loop on pixels in first (upper left) target MAPSel quadrant in zigzag order

Determine symmetric target-pixel zigzag addresses in other three quadrants

Accumulate convolution sums for target pixels from each quadrant for active elements from nine associated target/surround regions

Reset (smooth) target pixel from each quadrant with additive random dither if designated

Return for next MAPSel until subframe is exhausted

Output adaptively smoothed subframe

CALLED from: ADAPT

COMMON /WGTDIT/ KCW,SIG,DIT
COMMON /SFLOOP/ KSF,NSG,NSL,NPS,KLS,KPS
COMMON /GAZGIZ/ NZZ(1024)
COMMON /LPZZ/ LZZ(1024),NZZ(1024)
COMMON /WGTm/ WGT(9,340)
COMMON /SHAKE/ ISFD(65,34),ISFL(65,34)
BYTE ISFD,ISFL
COMMON /SFUDAIA/ ISFA(1024)
BYTE ISFA

DIMENSION LVSZ(5),MOSZ(5),MXZ(5),MSZ(5),MHSZ(5),LVWGT(5),IWT(9,4)
DIMENSION L5UR(4),KSUR(4),DSUR(16),SWR(16),LQZ(4),8W(4),SWI(4)
DATA LVSZ/4,16,64,256,1024/
DATA MDSZ/1,4,16,64,256/
DATA MAXS/3,15,63,255,1023/
DATA MSZ/2,4,8,16,32/
DATA MHSZ/1,2,4,8,16/
DATA LVWGT/0,0,4,20,84/
DATA IWT/1,2,3,5,6,7,9,10,11, 4,3,2,8,7,6,12,11,10, + 13,14,15,9,10,11,5,6,7, 16,15,14,12,11,10,8,7,6/
DATA IR,JE/0/
JZZ=1
100 L=LZZ(JZZ)
K=KZ(JZZ)
LV=ISFL(K,L)
IF(LLV,LV,0) LV=LV+256
LV=LV/32
IF(LV,E,1) GO TO 100
MAPS Adaptive Image Smoothing: TransMAPS 4-17

ADAPT/SFADPT

LSUR(1) = L-1
LSUR(2) = L
LSUR(3) = L + MHSZ(LV)
LSUR(4) = L + MSZ(LV)
KSUR(1) = K-1
KSUR(2) = K
KSUR(3) = K + MHSZ(LV)
KSUR(4) = K + MSZ(LV)
JT = 0
DO 120 LT = 1, 4
LS = LSUR(LT)
DO 110 KT = 1, 4
KS = KSUR(KT)
JT = JT + 1
IT = ISFL(KS, LS)
IF(IT.LT.0) IT = IT + 256
SUR(IJT) = IT
LVT = ISFL(KS, LS)
IF(LVT.LT.0) LVT = LVT + 256
SUR(JT) = IT
IF((LVT-256).GT.44) SURb(JT) = 0.
110 CONTINUE
120 CONTINUE

JWT = LV*GT(LV)
NQ = MSZ(LV)
DO 170 LUCZZ = 1, NQ
LOZZ(1) = LUCZZ - 1
LOZZ(4) = MSZ(LV) - LOZZ(1)
LOZZ(2) = LOZZ(1), AND ."1252", OR, (LOZZ(4), AND ."525")
LOZZ(3) = LOZZ(1), AND ."525", OR, (LOZZ(4), AND ."1252")
DO 130 JU = 1, 4
SW(JU) = 0.
130 SWI(JU) = 0.
JWT = JWT + 1
DO 150 J = 1, 9
WT = GT(J, JWT)
DO 140 JO = 1, 4
I = IMT(J, JO)
SWT = WT * SURL(I)
SW(JO) = SW(JO) + SWT
SWI(JO) = SWI(JO) + SWT * SURL(I)
140 CONTINUE
150 CONTINUE
DO 160 JO = 1, 4
I = IMT(5, JO)
AI = SURL(I)
IF(SW(JO), GT, 0.) AI = SWI(JO) / SW(JO)
IF(DIT, LE, 0.) AI = AI + DIT * (RAN(1R, JR) - .5)
IT = AI + 0.5
IF(IT.NE.0) IT = 0
IF(IT.GT.255) IT = 255
IF(IT.GT.127) IT = IT - 256
JJZZ = JJZ + LGZZ(JO)
IRST = ZZ(JJZZ)
160 CONTINUE
170 CONTINUE
GO TO 200

III-104
MAPS Adaptive Image Smoothing: TransMAPS 4-18

180  IRST=NZZ(JZZ)
     ISFA(IRST)=ISFD(K,L)
     IF(LV.EQ.0) GO TO 200
     DO 190 J=1,3
     JJZZ=JZZ+J
     L=LZZ(JJZZ)
     K=KZZ(JJZZ)
     IRST=NZZ(JJZZ)
190  ISFA(IRST)=ISFD(K,L)
     INCZZ=1
     IF(LV.NE.0) INCZZ=LVSZ(LV)
     JZZ=JZZ+INCZZ
     IF(JZZ.LE.MSQ) GO TO 100
     WRITE (4) (ISFA(J),J=1,NSQ)
     RETURN
     END

111-105
11.1 Program Characteristics

Program Names: DIFFER or DF

Subroutines: USERE
             FILESE
             SETUP
             SFDIFF
             ESTATS

Files:     EPRINT.DAT (listing)
           IMAGE.DAT or DMAPS.DAT (input)
           DMAPS.DAT or ADAPT.DAT (input)
           ERROR.DAT (output)

Task Build Options: MAXBUF = 1024
                    ACTFIL = 5

Task Size: 25312

11.2 Source Listing

The COMMENT-annotated source listing for DIFFER follows:
DIFFERENCE IMAGE AND STATISTICS FORMATION: TransMAPS 5-1

PROGRAM DIFFER

CALL USERE
CALL FILESE
CALL SETUPE
CALL SFDIFF
CALL SFDIFF

COMMON Block Communication:

/DIFF/ User image controls

/I1TYP(/) [Byte] Filename for input image #1
/I2TYP(/) [Byte] Filename for input image #2
/FAC [R*4] Difference image control parameter:
< 0 form signed and biased difference
= 0 find difference statistics only
> 0 form amplified absolute difference
FAC is amplification factor

/DIFFOP/ Derived difference controls

/SETUPE,SFDIFF

/FACT [R*4] Cop of FAC
/FS [R*4] Activation factor for signed difference
/FA [R*4] Activation factor for amplified difference
### Difference Image and Statistics Formation: TransMAPS 5-2 DIFFER

#### /HEAD1/ Standard MAPS file header, image #1

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF1</td>
<td>File type</td>
<td>[1*2]</td>
</tr>
<tr>
<td>INAME1</td>
<td>User-selected image name</td>
<td>[byte]</td>
</tr>
<tr>
<td>NL1</td>
<td>Number of lines in source image</td>
<td>[1*2]</td>
</tr>
<tr>
<td>NP1</td>
<td>Number of pixels in source image</td>
<td>[1*2]</td>
</tr>
<tr>
<td>NB1</td>
<td>Number of bits/pixel in source image</td>
<td>[1*2]</td>
</tr>
<tr>
<td>KS1</td>
<td>Kind of subframe 8x8 16x16 32x32</td>
<td>[1*2]</td>
</tr>
<tr>
<td>IGRD1</td>
<td>Subframe grid: square(0)/staggered(1)</td>
<td>[1*2]</td>
</tr>
<tr>
<td>MS1</td>
<td>Total subframe count</td>
<td>[1*4]</td>
</tr>
<tr>
<td>MC1</td>
<td>MAPSel count</td>
<td>[1*4]</td>
</tr>
<tr>
<td>NISBP1</td>
<td>Packed block(0)/pattern(1) mode (rt-lft)</td>
<td>[1*2]</td>
</tr>
<tr>
<td>IBV1(5,2)</td>
<td>Optimal pattern biases by level,low/high</td>
<td>[1*2]</td>
</tr>
<tr>
<td>IPAD1(7)</td>
<td>Space for future extension</td>
<td></td>
</tr>
</tbody>
</table>

#### /HEAD2/ Standard MAPS file header, image #2

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF2</td>
<td>File type</td>
<td>[1*2]</td>
</tr>
<tr>
<td>INAME2</td>
<td>User-selected image name</td>
<td>[byte]</td>
</tr>
<tr>
<td>NL2</td>
<td>Number of lines in source image</td>
<td>[1*2]</td>
</tr>
<tr>
<td>NP2</td>
<td>Number of pixels in source image</td>
<td>[1*2]</td>
</tr>
<tr>
<td>NB2</td>
<td>Number of bits/pixel in source image</td>
<td>[1*2]</td>
</tr>
<tr>
<td>KS2</td>
<td>Kind of subframe 8x8 16x16 32x32</td>
<td>[1*2]</td>
</tr>
<tr>
<td>IGRD2</td>
<td>Subframe grid: square(0)/staggered(1)</td>
<td>[1*2]</td>
</tr>
<tr>
<td>MS2</td>
<td>Total subframe count</td>
<td>[1*4]</td>
</tr>
<tr>
<td>MC2</td>
<td>MAPSel count</td>
<td>[1*4]</td>
</tr>
<tr>
<td>NISBP2</td>
<td>Packed block(0)/pattern(1) mode (rt-lft)</td>
<td>[1*2]</td>
</tr>
<tr>
<td>IBV2(5,2)</td>
<td>Optimal pattern biases by level,low/high</td>
<td>[1*2]</td>
</tr>
<tr>
<td>IPAD2(7)</td>
<td>Space for future extension</td>
<td></td>
</tr>
</tbody>
</table>

#### /SFDAT1/ Subframe data, input image #1

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISF1(1024)</td>
<td>Image #1 subframe data array</td>
<td>[byte]</td>
</tr>
</tbody>
</table>

#### /SFDAT2/ Subframe data, input image #2

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISF2(1024)</td>
<td>Image #2 subframe data array</td>
<td>[byte]</td>
</tr>
</tbody>
</table>

#### /SFDAT3/ Subframe data, difference image

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSG</td>
<td>Number of total pixels/subframe</td>
<td>[1*2]</td>
</tr>
<tr>
<td>ISF3(1024)</td>
<td>Difference image subframe data array</td>
<td>[byte]</td>
</tr>
</tbody>
</table>

#### /STATSE/ Image distributions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1VS12(64,64)</td>
<td>Image #1 vs Image #2 2-D histogram</td>
<td>[1*4]</td>
</tr>
<tr>
<td>I1H(64)</td>
<td>Image #1 histogram</td>
<td>[1*4]</td>
</tr>
<tr>
<td>I2H(64)</td>
<td>Image #2 histogram</td>
<td>[1*4]</td>
</tr>
<tr>
<td>I1M2H(64)</td>
<td>Difference histogram (All distributions in steps of 4 gray-scale level increments)</td>
<td>[1*4]</td>
</tr>
</tbody>
</table>

---

III-108
Differencing Image and Statistics Formation: TransMAPS 5-3

Fidelity measure accumulations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIX</td>
<td>Total pixel count for subframes</td>
</tr>
<tr>
<td>SUM1</td>
<td>Sum of image 1 gray-scale values</td>
</tr>
<tr>
<td>SUMSQU1</td>
<td>Sum of squares of image 1 values</td>
</tr>
<tr>
<td>DIFF</td>
<td>Sum of difference gray-scale values</td>
</tr>
<tr>
<td>DIFFSQ</td>
<td>Sum of squares of difference values</td>
</tr>
</tbody>
</table>

Common /HEAD1/ FILE1,INAME1(8),NL1,NP1,NB1,KSF1,IGRD1,NS1,MCI.
+ MIXBP1,IV1(5,2),IPAD1(7)
BYTE INAME1
INTEGER*4 NS1,MCI
COMMON /HEAD2/ FILE2,INAME2(8),NL2,NP2,NB2,KSF2,IGRD2,NS2,MC2.
+ MIXBP2,IV2(5,2),IPAD2(7)
BYTE INAME2
INTEGER*4 NS2,MC2
COMMON /DIFF/ I1ITYP(6),I2ITYP(6),FAC
BYTE I1ITYP,I2ITYP
INTEGER*4 LOUPSF,ISTAR4
DIMENSION FNAMES(6,4)
BYTE FNAMES
DATA FNAMES /IHI,INM,INH,1HE,1H ,1HD,1HM,1HA,1HP,1HS,1H ,
+ 1HL,1HE,1HV,1HE,1HL,1H ,1HA,1HD,1HA,1HP,1HT,1H /
CALL USER
CALL FILESE
II=FILE1
IF(I1.EQ.0) II=1
I2=FILE2
IF(I2.EQ.0) I2=1
TYPE 500,INAME1,(FNAMES(J,I1),J=1,5),INAME2,(FNAMES(J,I2),J=1,5)
500 FORMAT(/,1X,"DIFFERENCING ",0A1," TYPE ",5A1,
+ " VS ",0A1,
+ " TYPE ",5A1)
CALL SETUP
DO 110 LOOPSF=1,NS1
110 CALL SFDIFF
BPP1=NB1
IF(MC1.EQ.0) GO TO 210
ISTAR4=(MC1+2)/3
IF(KSF1.EQ.0) ISTAR4=(MC1+3)/4
ISTAR4=8*(MC1+ISTAR4)
DEN=L1
BPP1=ISTAR4/DEN
BPP1=BPP1/NP1
210 WRITE (1,600) INAME1,(FNAMES(J,II),J=1,5),BPP1
600 FORMAT(1H1,5X,"DIFFERENCE STATISTICS:",/10X,"IMAGE1 ",0A1,
+ ", TYPE ",5A1," BITS/PIXEL",F8.5)
BPP2=MB2
IF(MC2.EQ.0) GO TO 220
ISTAR4=(MC2+2)/3
IF(KSF2.EQ.0) ISTAR4=(MC2+3)/4
ISTAR4=8*(MC2+ISTAR4)
DEN=L2
BPP2=ISTAR4/DEN
BPP2=BPP2/NP2

111-109
Difference Image and Statistics Formation: TransMAPS 5-4

220 WRITE (1,610) IANAME2, (FNAME3(J,I2), J=1.5), BPP2
610 FORMAT(15X,"V5",/10X,"IMAGE2: ",BA1," ", TYPE: ",BA1,
", BITS/PIXEL: ",F6.5,/
IF(FAC.LT.0.) WRITE (1,620) MS1
620 FORMAT(15X,"SIGNED DIFFERENCE IMAGE PRODUCED: ",I7,
" SUBFRAMES",/)
IF(FAC.GT.0.) WRITE (1,630) MS1,FAC
630 FORMAT(15X,"AMPLIFIED DIFFERENCE IMAGE PRODUCED: ",I7,
" SUBFRAMES, AMPLIFICATION",F6.1,/)
CALL ESTATS
CLOSE(UNIT=1)
CLOSE(UNIT=2)
CLOSE(UNIT=3)
IF(FAC.NE.0.) CLOSE(UNIT=4)
END
SUBROUTINE USERE

C C Purpose: User interaction for difference image and fidelity performance evaluation
C Subframe image type for input image #1 (IMAGE or DMAPS)
C Difference control parameter
C
C CALLED from: DIFFER
C
COMMON /DIFF/ IITYP(6),I2TYP(b),FAC
BYTE IITYP,II2TYP
DIMENSION NAMES(6,3)
BYTE NAMES
DATA IITYP/IHI,1HM,1HA,1HE,0/
DATA I2TYP/IHD,1HM,1HA,1HE,0/
DATA FAC/10./
DATA NAMES/IHI,1HM,1HA,1HE,0,1HD,1HM,1HA,1HE,0,
+ 1HD,1HM,1HA,1HE,0/  
11=1
12=2
TYPE 500
500 FORMAT(/,1X,32(1H*),/1X,",",30X,"",/1X,
+ "MAPS DIFFERENCE IMAGE MODULE ",
+ /,1X,",",30X,"",/1X,32(1H*)))
100 TYPE 510
510 FORMAT(/,3X,"INPUT IMAGE TYPES:",/7X,
+ "I = IMAGE (ORIGINAL SOURCE) IMAGE1 ONLY",/7X,
+ "D = DMAPS (MAPS DECOMPRESSED)",/7X,
+ "A = ADAPT (ADAPTIVELY SMOOTHED) IMAGE2 ONLY")
TYPE 520,511YP
520 FORMAT(/,5X,"IMAGE1? (I D) "","A1","{","4A1","}","A1")
ACCEPT 1,LIT
1 FORMAT(10A1)
IF(LIT.EQ.,1HI) 11=1
IF(LIT.EQ.,1HD) 11=2
DO 110 11=1,6
110 IITYP(I)=NAMES(1,11)
IF(11.NE.2) GO TO 130
12=3
DO 120 12=1,6
120 I2TYP(I)=NAMES(1,12)
130 TYPE 530,12TYP
530 FORMAT(5X,"IMAGE2? (D A) "","A1","{","4A1","}","A1")
IF(11.EQ.2) GO TO 150
ACCEPT 1,LIT
IF(LIT.EQ.,1HD) 12=2
IF(LIT.EQ.,1HA) 12=3
DO 140 12=1,6
140 I2TYP(I)=NAMES(1,12)
Difference Image and Statistics Formation: TransMAPS 5-6

150 TYPE 540,FAC
540 FORMAT(/,3X,"DIFFERENCE IMAGE TYPE:","/",7X,
+ "AMPLIFICATION FACTOR","/",9X,
+ "<0 SIGNED AND BIASED DIFFERENCE IMAGE","/",9X,
+ "=0 NO DIFFERENCE IMAGE, STATISTICS ONLY","/",9X,
+ ">0 AMPLIFIED DIFFERENCE IMAGE","/",15X,
+ "VALUE IS AMPLIFICATION","/",5X,
+ "FACTOR?",F5.0,5X,"(/ = NO CHNG)"
+ ACCEPT *,FAC
TYPE 550
550 FORMAT(/,3X,"USER SPECIFICATION COMPLETE:","/",3X,28(1H*),"/",5X,
+ "REVIEW? (Y OK N) N"
+ ACCEPT 1,LIT
IF(LIT.EQ.1MY) GO TO 100
RETURN
END
SUBROUTINE FILESE

Purpose: Open files, check integrity, and read/write file headers

CALled from: DIFFER

COMMON /HEAD1/ IFILEI,INAME1(8),ML1,NP1,NSI,KSF1,IGRD1,NS1,MC1,
+ BYTE INAME1
INTEGER*4 NS1,MC1
DIMENSION JHEAD1(32)
EQUVALENCE (JHEAD1(1),IFILE1)
COMMON /HEAD2/ IFILE2,INAME2(9),ML2,NP2,NS2,KSF2,IGRD2,NS2,MC2,
+ BYTE INAME2
INTEGER*4 NS2,MC2
DIMENSION JHEAD2(32)
EQUVALENCE (JHEAD2(1),IFILE2)
COMMON /DIFF/ 11TYP6),12TYP6),FAC
BYTE 11TYP,12TYP
COMMON /SFDAT/ NSQ*1SF3C1024)
BYTE 1SF3
DIMENSION JSF(512)
EQUVALENCE (JSF(1),ISF3(1))
OPEN(UNIT=2,TYPE="OLD",NAME=11TYP,FOR=UNFORMATTED',
+ RECORDTYPE="FIXED")
READ (2) JHEAD1
11=IFILE1
IF((11.EQ.0),OR,((11.GE.2),AND,(11.LE.4))) GO TO 110
IT1=1
500 TYPE 500,IT1
FORMAT("/X,** FILE",I2," TYPE",I2," INVALID INPUT'")
STOP 1
110 OPEN(UNIT=3,TYPE="OLD",NAME=12TYP,FOR=UNFORMATTED',
+ RECORDTYPE="FIXED")
READ (3) JHEAD2
12=IFILE2
IF((12.EQ.0),OR,((12.GE.2),AND,(12.LE.4))) GO TO 120
IT2=2
500 TYPE 500,IT2
STOP 2
120 IF(KSF1.EQ.KSF2) GO TO 130
510 FORMAT("/X,** SUBFRAME SIZES DISAGREE",I3," VS",I3)
STOP 130 IF(NS1.EQ.NS2) GO TO 200
520 FORMAT("/X,** SUBFRAME COUNTS DISAGREE",I7," VS",I7)
STOP

III-113
Difference Image and Statistics Formation: TransMAPS 5-8

200 NSQ=KSF1*KSF1
   IF(FAC.EQ.0.) GO TO 250
   LSQ=NSQ/4
   OPEN(UNIT=4,TYPE="NEW",NAME="ERROR",FORM="UNFORMATTED",
        RECORDTYPE="FIXED",RECORDSIZE=LSQ)
   DO 210 J=1,32
210 JSF(J)=JHEAD1(J)
   IFILE3=5
   IF(I2TYPE(1).EQ.1HA) IFILE3=6
   IF(I1TYPE(1).EQ.1HD) IFILE3=7
   JSF(1)=IFILE3
   WRITE (4) (JSF(J),J=1,NSQ)
   DD 220 J=1,32
220 JSF(J)=0
250 OPEN(UNIT=1,TYPE="NEW",NAME="EPRINT")
   RETURN
END
SUBROUTINE SETUPE

C Purpose: Initialize statistical sums and accumulation arrays

C CALLED from: DIFFER

C-----------------------------------------------
COMMON /DIFF/ ITYP(6),12TYP(6),FAC
BYTE ITYP,12TYP
COMMON /DIFFOP/ FACT,FS,FA
COMMON /SIMS/ PX,SUM1,SUMSQ1,DIFF,DIFFSQ
REAL*8 PIX,SUM1,SUMSQ1,DIFF,DIFFSQ
COMMON /STATSE/ IIVS12(64,64),I1H(64),I2H(64),I1M2H(64)
INTEGER*4 IIVS12,I1H,I2H,I1M2H
FACT=1
FS=1.
FA=0.
IF(FACT.LE.0.) GO TO 110
FS=0.
FA=FACT
CONTINUE
PIX=O.DO
SUM1=O.DO
SUMSQ1=O.DO
DIFF=O.DO
DIFFSQ=O.DO
DO 130 J=1,64
   DO 120 I=1,64
      IIVS12(I,J)=O
      I1H(J)=O
      I2H(J)=O
   120   I1M2H(J)=O
130   RETURN
END

III-115
Difference Image and Statistics Formation: TransMAPS 5-10  DIFFER//SFDIFF

SUBROUTINE SFDIFF

C
Purpose: Accumulate difference statistics and generate difference
image subframe
CALLED from: DIFFER

COMMON /DIFFOP/ ACT,FS,FA
COMMON /SUMS/ PIX,SUM1,SUMSQ1,DIFF,DIFFSQ
REAL*8 PIX,SUM1,SUMSQ1,DIFF,DIFFSQ
COMMON /STATS/ 1113,1,11M2,11H(64),12H(64),11M2H(64)
INTEGER*4 1113,11H,12H,11M2H
COMMON /SFDAT1/ ISF1(1024)
BYTE ISF1
COMMON /SFDAT2/ ISF2(1024)
BYTE ISF2
COMMON /SFDAT3/ NSQ,ISF3(1024)
BYTE ISF3
READ (2) (ISF1(J),J=1,NSQ)
READ (3) (ISF2(J),J=1,NSQ)
DO 110 I=1,NSQ
ISF1(I)
IF(I,I.IT.0) I1=1+256
I2=ISF2(I)
IF(I,I.IT.0) I2=12+256
PIX=PIX+1
DO SUMIXSUM1+GS1
SUMSO1BSUNSG1,GS1 SGSI
DIFF=DIFF+GSIM2
DIFFS=DIFFS+GS1M2*GSI N2
J1=I1/4+1
J2=12/4+1
J1M2=I1M2/4+33
IF(J1M2.GT.64) J1M2=64
1113=1113(J1,J2)*1113(1113,J2)+1
11H(J1)=11H(J1)+1
12H(J2)=12H(J2)+1
11M2H(J1M2)=11M2H(J1M2)+1
IPOB=IIM2
IF(IPOB.LT.0) IPOB=IPOB
IDIFF=FS*(11M2+127)+FA*IPDS
IF(IDIFF.LE.0) IDIFF=0
IF(IDIFF.GT.255) IDIFF=255
IF(IDIFF.GT.127) IDIFF=IDIFF-256
ISF3(I)=IDIFF
110 CONTINUE
IF(FACT.E.0.) WRITE (4) (ISF3(J),J=1,NSQ)
RETURN
END

III-116
Purpose. Output fidelity measures and image distributions

CALLED from: DIFFER

COMMON /SUMS/ PIX, SUM1, SUMSQ1, DIFF, DIFFSQ
REAL*8 PIX, SUM1, SUMSQ1, DIFF, DIFFSQ
COMMON /STATS/ 11V512(64,64), I1H(64), I2H(64), I1M2H(64)
INTEGER*4 11V512, I1H, I2H, I1M2H
DIMENSION LBLL(4), LINE(64), LSYM(11)
INTEGER*4 NSWPIX, ISTAR4
REAL*8 SSQREL
DATA LBBLL/29*I H, 1H, 1H, I1A, 1H, I1E, 1H, I1M2, 28*I H /
DATA LSYM/1H, 1H, 1H, 1H, 1H, 1H, 1H, 1H, 1H, 1H, 1H /
AVDIFF=DIFF/PIX
RATIO=DIFFSQ/PIX
RSSQ=SQRT(RATIO)
ERR=0.
IF(SUMSQ1 GT 0.0) ERR=100.*((DIFFSQ/SUMSQ1)
REL=0.
SSQREL=SUMSQ1-SUM1*SUM1/PIX
IF(SSQREL GT 0.0) REL=100.*((DIFFSQ/SSQREL)
WRITE (1,640) AVDIFF, RMS, ERR, REL
640 FORMAT(10X,"DIFFERENCES, IMAGE1-IMAGE2:
MEAN","F8.3"," RMS","F9.5","% RMS","F9.5","%",/)
WRITE (1,640) NSWPIX, PIX
700 FORMAT(20X,"IMAGE 1 (UNIT","16"," PIXELS)",40X,
+ "DISTRIBUTION",40X."GRAY SCALE IMAGE 1 IMAGE 2",4X,
+ "DIFFERENCE IMAGE1-IMAGE2",4X,
+ 64(1H=","+")
DO 150 J=1,64
   DO 110 I=1,64
      LINE(I)=I1H
      LINE(J)=I2H
      DU 120 I=1,64
      ISTAR4=11V512(I,J)
      IF(ISTAR4, EQ, 0) GO TO 120
      I1=ISTAR4/NSWPIX+1
      IF(I1, GT, 10) I1=11
      LINE(I)=LSYM(I1)
   110 CONTINUE
   WRITE (1,710) LBLL(J), LINE, LU, I1H(J), I2H(J), LDL, LDU, I1M2H(J)
120 CONTINUE
WRITE (1,710) LBLL(J), LINE, LU, I1H(J), I2H(J), LDL, LDU, I1M2H(J)
710 FORMAT(1X,AL, "","", 64AL, ",", 14, " TO", 14, ",", 14, " TO", 14, ",", 14, " TO", 15, ",", 11)
150 CONTINUE
WRITE (1,720)
720 FORMAT(3X","","64(1H=","+")
RETURN
END
SECTION TWELVE

TRANSMAPS MODULE #6: SUBFRAME TO RASTER CONVERSION

12.1 Program Characteristics

Program Names: RASTER or RS

Subroutines: USERR
             FILESRR
             SETURR
             UNSQR
             UNSTGR

Files: IMAGE.DAT (input)
       or DMAPS.DAT
       or LEVEL.DAT
       or ADAPT.DAT
       or ERROR.DAT
       or IRAST.DAT (output)
       or DRAST.DAT
       or LRAST.DAT
       or ARAST.DAT
       or ERAST.DAT

Task Build Options: MAXBUF = 1024

Task Size: 29664

12.2 Source Listing

The COMMENT-annotated source listing for RASTER follows:
Subframe to Raster Conversion: TransMAPS 6-1

PROGRAM RASTER

*-------------------------------------------------------------------*
| TransMAPS Module: 6: Subframe to Raster Conversion                |
*-------------------------------------------------------------------*

Control Data Corporation - 1982

<table>
<thead>
<tr>
<th>Files: Unit</th>
<th>Name</th>
<th>Content</th>
<th>From/To Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>In</td>
<td>2</td>
<td>IMAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Source image, subframes</td>
<td>SUBFRM DIRECT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or DMAPS Decompressed image, subframes</td>
<td>DMAPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or LEVEL Resolution image, subframes</td>
<td>DMAPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or ADAPT Adaptively smoothed, subframes</td>
<td>ADAPT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or ERROR Difference image, subframes</td>
<td>DIFFER</td>
</tr>
<tr>
<td>Out</td>
<td>3</td>
<td>IRAST</td>
<td>User SEGMENTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Source image, raster</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>or DRAST Decompressed image, raster</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>or LRAST Resolution image, raster</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>or ARAST Adaptively smoothed, raster</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>or ERAST Difference image, raster</td>
<td></td>
</tr>
</tbody>
</table>

User Interaction: In Subroutine USERR

Image Type Selection for Conversion

Program Structure:

PROGRAM RASTER

CALL USERR Specify image to be converted
CALL FILES Open files, read header, check type integrity
CALL SETUP Establish partition controls, check image size integrity
IF square grid:
   CALL USQR Convert 4x8 16x16 32x32 subframes to raster
      Loop on rows of subframes in line direction
      Loop on 8-line swaths within rows
      Loop on subframes in pixel direction
      Input subframe
      Extract line segments for swath
      Output completed lines (burst of 8)
   or IF staggered grid:
   CALL UNSQR Convert 4x8 staggered subframes to raster
      Preload buffer with all subframes with first line
      Loop on lines
      Output current line
      Replace all subframes needed to finish next line

COMMON Block Communication:

Blank, Raster image data Length: 16000
RASTER, USQR, UNSQR

ISUF(4000,0) (Byte) Block or recirculating 8-line buffer

III-119
Subframe to Raster Conversion: TransMAPS 6-2

/DSPR/  Square grid partition controls  Length: 3
SETUPR, UNSOR

WPS  [I+2]  Number of pixel-direction subframes
NL5  [I+2]  Number of line-direction subframes
NWT  [I+2]  Number of staggered/subframe (1 2 4)

/DSGR/  Staggered grid partition controls  Length: 17
SETUPR, UNESCO

NPST  [I+2]  Number of pixel-direction subframes total
NPST(0)  [I+2]  Number of staggered subframes in pixel
direction for each startline mod 8
IPST(0)  [I+2]  Initial pixel of first staggered
subframe for each startline mod 8

/HEADER/  Standard MAPS file header  Length: 32
MASTER, FILESR, SETUPR

.IFILE  [I+2]  File type
INAME(0)  [Byte]  User-selected image name
ML  [I+2]  Number of lines in source image
NP  [I+2]  Number of pixels in source image
NB  [I+2]  Number of bits/pixel in source image
KSF  [I+2]  Kind of subframe 6x8 16x16 32x32
IGRD  [I+2]  Subframe grid: square(0)/staggered(1)
MS  [I+4]  Total subframe count
MC  [I+4]  MAPSet count
MKB  [I+2]  Packed block(0)/pattern(1) mode (rt-lst)
IDV(5,2)  [I+2]  Optimal pattern biases by level,low/high
IPAD(7)  [I+2]  Space for future extension

/OUTNR/  Output file identification  Length: 3
MASTER, USERNR, FILESR

NAME(6)  [Byte]  Output file name

/SFDATA/  Subframe data  Length: 512
UNSOR, UNESCO

ISF(1024)  [Byte]  Subframe data input array

/SFTEMP/  Subframe control parameters  Length: 4
SETUPR, UNSOR, UNESCO

MLT  [I+2]  Number of lines in image
NPT  [I+2]  Number of pixels in image
KBST  [I+2]  Subframe size: edge
KSST  [I+2]  Subframe size: pixel count

/SUBFNM/  Input file identification  Length: 3
USERR, FILESR

NAMEF(6)  [Byte]  Input file name

III-120
COMMON IBUF(4000,9)
BYTE IBUF
COMMON /HEADER/ IFILE,IRNAME(5),NL,MP,MR,RSF,ICMD,NS,NC,IVRNP.
  IBVS(5,2),IPAD(7)
BYTE IRNAME
INTEGER NS,NC
COMMON /OUTER/ RNAME(6)
BYTE RNAME
DIMENSION RNAME(9,9)
BYTE RNAME
DATA RNAME/IN,IN,IN,IN,IN,IN,IN,IN,IN
  IN,IN,IN,IN,IN,IN,IN,IN,IN
  IN,IN,IN,IN,IN,IN,IN,IN,IN
  IN,IN,IN,IN,IN,IN,IN,IN,IN
  IN,IN,IN,IN,IN,IN,IN,IN,IN
  IN,IN,IN,IN,IN,IN,IN,IN,IN
  IN,IN,IN,IN,IN,IN,IN,IN,IN
  IN,IN,IN,IN,IN,IN,IN,IN,IN
  IN,IN,IN,IN,IN,IN,IN,IN,IN
  IN,IN,IN,IN,IN,IN,IN,IN,IN
CALL USER
CALL FILEIN
CALL SETUP
ZFILE=IFILE+1
TYPE 500,IRNAME(IMPORTS,JFILE),J=1,9),NL,MP,RNAME
FORMAT(/,1X,"CONVERTING IMAGE ",JFILE"," FILE TYPE: ",JFILE,/, 
IF(IORD,NE,0) GO TO 110
CALL USER
GO TO 120
110 CALL USER
120 CONTINUE
CLOSE(UNIT=2)
CLOSE(UNIT=3)
END
SUBROUTINE USERP

C	Purpose: User interaction for submenu to raster conversion
   Image type to be converted

C	Called from: USER

COMMON /SUBPWR/ NAMEF(6)
COMMON /OUT0M/ NAMEP(6)
COMMON /OUT1M/ NAMEP(6)

DATA NAMEF/IMN,1MA,1MP,1NS,0/
DATA NAMEP/IMN,1MA,1MP,1NS,INT,0/
DATA NAMEP/IMN,1MA,1NS,INT,0,1MP,1NS,0,
DATA NAMEP/IMN,1MA,1NS,INT,0,1MP,1NS,0,
DATA NAMEP/IMN,1MA,INT,0,1NS,INT,0,
DATA NAMEP/IMN,1MA,INT,0,1NS,INT,0,

OUT=3	1 Default

TYPE 500
900 FORMAT(/12,45(1X),/12,",","",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",","
SUBROUTINE FILEEO

PURPOSE: Open files, read header, check type integrity

CALLED FROM RASTER

COMMON /NAMEO/, IFILE, IMAX(1), ML, MP, MO, MF, IFIO, OS, MC, OIlBDP,
  IRJY(5,3), IMAGE(7)
BITE IMAGE
INTEGER AO, OF
DIMENSION JREAD(12)
COMMON /NAMEO/, IFILE
BITE NAME
COMMON /NAMEO/, NAME(16)
BITE NAME
OPENUNIT=0, TYPE= "O", NAME= NAME, FOR= "ACCESS= "DIRECT"
READ (1211) JREAD
IF (IFILE.EQ.0) GO TO 666 IFILE.EQ.2 IFILE.EQ.717)
GO TO 110
OPENUNIT=0, TYPE= "O", NAME= NAME, FOR= "ACCESS= "DIRECT"
GO TO 120

110 TYPE IFILE, IMAGE
666 FORMAT(18, "ERROR INVALI0D FILE TYPE: IFILE =", 10.
   ", IMAGE =", 011)
120 RETURN
END
SUBROUTINE RASTER

PURPOSE: Convert single-grid raster to raster image

VARIABLES: RASTER, GRID, ALL

CONVERTING SEQUENCE:
1. LOAD RASTER
2. OUTPUT CURRENT LINE
3. OUTPUT ALL CONSECUTIVE LINES

CALLED FROM RASTER

CPU TIKS

00000 00000 00000

00000 00000 00000

00000 00000 00000

00000 00000 00000

00000 00000 00000

00000 00000 00000

00000 00000 00000

00000 00000 00000

00000 00000 00000
SUBROUTINE TO Raster Conversion: TRANSDIR 0-9

IF(LGE, 41, LQ) GO TO 300
JPS1=JPS1+1
IF(JPS1.LT.1) GO TO 210
JPS2=JPS2+1
JPS3=JPS3+1
IF(JPS3.LT.1) GO TO 100
DO 200 JPS1=JPS1,JPS2-1
READ (L3,J3C(1)) JPS1,JPS2,JPS3
JPS1=JPS1+1
100 CONTINUE
DO 300 JPS1=JPS1,JPS2-1
READ (L3,J3C(1)) JPS1,JPS2,JPS3
JPS1=JPS1+1
200 CONTINUE
300 CONTINUE
100 RETURN
END
SECTION THIRTEEN

TRANSMAPS MODULE 07: IMAGE ASSEMBLY AND ANNOTATION

13.1 Program Characteristics

Program Names: AMODE or AI

Subroutines: USER

Files: SUNQD.DAT (input)
        USER1.DAT (input raster image)
        USER2.DAT (input raster image)
        AMODE.DAT (output)
        or AMODE.LST (listing)

Task Build Options: 

Task Size: 31000

13.2 Source Listing

The COMPARE-annotated source listing for AMODE follows:
PROGRAM ANNOT

[Program flowchart and annotations]

Control Data Corporation - 1967

Output Image Specifications:

[Annotations and structure diagrams related to output image specifications]

Program Structure:

[Program flowchart and annotations]

III-129
COMMON Block Communication.

Embedded Image Specification

Length: 31

BING (1) Number of Embedded Images (0=2)
FILENAM (10,2) User input file names
JL (2) Input lines to skip
JP (2) Input pixels to skip
SL (2) Input lines to transfer
SP (2) Input pixels to transfer
ISP (2) Total input pixels (skip + transfer)
IL (2) Initial (starting) line on output image
IL (2) Last line on output image
IP (2) Initial (starting) pixel on output image
LP (2) Last pixel on output image
IC (2) Input image complement select (non-0)

Annotation Specification

Length: 641

NAGS (1) Number of Messages (0-20)
AAGO (20) Message orientation (TOLA)
AGM (20) Message character count (50 maximum)
AGM (20) Message symbol size (1x 1x 1x 1x)
AGM (20) Symbol size in pixels
AGM (20) Message center pixel on output image
AGM (20) Message complement select (non-0)
TEST (20,20) Message test

Embedded Image/Message Assembly

Length: 200

LTP (4000) Temporary staging array

Local Message Control Parameters

Length: 44

SY (1) Message symbol size
NL (1) Line within message frame
ML (1) Number of lines/columns from symbol map
ML (1) Indents of lines/columns from symbol map
CT (1) Current symbol index
LF (1) Extracted lines/columns from symbol map
CH (16) Annotation pixels for this symbol & line

Message Line and Pixel Ranges

Length: 80

NIL (20) Message initial line on output image
NLL (20) Message last line on output image
NLP (20) Message initial pixel on output image
NLP (20) Message last pixel on output image

111-120
Annotation and Image Assembly: TransHAPS 7-3

/RODEG/ Output Image mode
   ANNOTE,USER,FILSET
   Length: 1

/PANT/ [2] Raster image(0)/Printer pseudo-image(1)

/OUTING/ Output Image specification
   ANNOTE,USER,FILSET,OVERLAP
   Length: 3

/ELI [2] Number of lines in Output Image
/SP [2] Number of pixels in Output Image
/BOAP [2] Background complement select (non-0)

/OUTLIN/ Output line assembly
   ANNOTE,FILSET
   Length: 2000

/LINE(4000) [Byte] Line array

/SYMBOL/ Symbol map table
   61=BIT,LINES1,CLASS1
   Length: 8640

/STMB(3,40,60) [6] 3 words/line 40 lines/symbol 60 symbols
   (bit map packed left-to-right in word)

Orientation conventions:

A CDP-based coordinate convention is used:
   Origin is in the upper left corner
   The pixel index advances to the right (pixel direction)
   The line index advances downward (line direction)

Pixel Direction
   --------------
   |  Top |  
   |     |  
   |     |  
   -----

Line Direction
   -------
   |  Bottom |  
   |       |  
   |       |  

III-131
CALL USER
CALL SINDEX
CALL SYNBIN
CALL FILESET
IF(IPRNT.EQ.0) TYPE 500, NL, NP
500 FORMAT(/,IX,"ASSEMBLING AND ANNOTATING IMAGES",/,
      + 5X,IS," LINES BY", IS, " PENCES TO FILE "ANIMG.DAT")
IF(IPRNT.NE.0) TYPE 501, NL, NP
501 FORMAT(/,IX,"ASSEMBLING AND ANNOTATING IMAGES",/,
      + 5X,IS," LINES BY", IS, " PENCES TO FILE "APRINT.DAT")
   JCOL=128
   IF(NP.LT.128) JCOL=NP
   JCOL=JCOL+1
   JSYMP=1H*
   JSYMM=1H*
   IF(IPRNT.NE.0) WRITE (4, 510) (JSYMM, J=1, JCOL), JSYMP
510 FORMAT(1H1,",",129A1)
   DO 400 L=1, NL
   400 DO 210 J=1, NP
   210 LINE(J)=0
   IF(KOMP.EQ.0) GO TO 230
   DO 220 J=1, NP
   220 LINE(J)="777"
   230 CONTINUE
   IF(NIMG.LE.0) GO TO 350
   DO 330 I=1, NIMG
   IF(L.LT.IIL(I)) GO TO 330
   IF(L.GT.IL(I)) GO TO 330
   IUM=I+1
   INUMP(I)
   READ (1U, END=320) (LTMP(J), J=1, IN)
   JI=IP(I)
   JF=LP(I)
   JJ=JP(I)
   JC=IC(I)
   DO 310 J=JI, JF
   310 CONTINUE
   CONTINUE
   CLOSE(UNIT=IU)
   LL(I)=L-1
   LI=LL(I)
   KL(I)=LI
   LL(I)=JI
   KJ=JJ+1
   IF(JC.EQ.0) LTMP(JJ)=.NOT.LTMP(JJ)
   LINE(J)=LTMP(JJ)
   320 CONTINUE
   CONTINUE
   DO 330 I=1, NIMG
   IF(L.LT.IIL(I)) GO TO 330
   IF(L.GT.IL(I)) GO TO 330
   IUM=I+1
   INUMP(I)
   READ (1U, END=320) (LTMP(J), J=1, IN)
   JI=IP(I)
   JF=LP(I)
   JJ=JP(I)
   JC=IC(I)
   CONTINUE
   CONTINUE
   CALL TTSSETL
IF(IPRNT.EQ.0) GO TO 390
DO 380 J=1,JCOL
   LJ=LINE(J)
   IF(LJ.LT.0) LJ=LJ+256
   OLD=LJ/32+1
   LINE(J)=ISYM1(LJ)
   LTMP(J)=ISYM2(LJ)
380 CONTINUE
   LINE(JCOL1)=1H1
   WRITE (4,520) (LINE(J),J=1,JCOL1)
520 FORMAT(1X,"*",129A1)
   WRITE (4,530) (LTMP(J),J=1,JCOL)
530 FORMAT(1X,"*",128A1)
   GO TO 400
390 WRITE (4) (LINE(J),J=1,NP)
400 CONTINUE
IF(IPRNT.NE.0) WRITE (4,540) (JSYMP,J=1,JCOL),JSYMP
540 FORMAT(1X,"*",129A1)
ENDIF 4
IF(MIN.GE.1) CLOSE(UNIT=2)
IF(MIN.GE.2) CLOSE(UNIT=3)
CLOSE(UNIT=4)
END
SUBROUTINE USEN

Purpose: User interaction for image assembly
Output image specification
Model: raster image file or line-printer pseudo-image
Frame size and background direct/complement select
Embedded image specification
Image count (0-12)
Image description
File identification
Input positioning (line & pixel skips)
Size (line & pixel counts)
Output location (starting line and pixel)
Direct/complement select
Embedded annotation
Message count (0-20)
Message description
Message overlap assessment and editing

Called from: ANNOTE

Calls: OVHLAP, UMSG

Common /OUTING/ NL, NP, KOMP
Common /EMBIMG/ NIMG, FILEMAN(10,2), JL(2), JP(2), KL(2), KP(2),
+ INP(2), LL(2), IP(2), LP(2), IC(2)
BYTE FILEMAN
Common /EMBMSG/ NMSG, MSGO(20), MSGM(20), MSGN(20), M16(20),
+ MSGC(20), TEXT(30,20)
BYTE TEXT
Common /BOUND/ MIL(20), MLL(20), MPH(20), MLP(20)
Common /MODEU4/ IPRNT
Dimension NAMET(10)
BYTE NAMET, NAME1
Equivalence (NAME1, NAMET(1))
Dimension LBLG(4), MSGT(80)
BYTE LBLG, MSGT
Data MPIX, IPIX/4000, 4000/

--- Defaults ---
Data IPRNT/O/
Data NL, NP/800, 800/
Data KOMP/O/
Data NIMG/O/
Data FILEMAN/IHT, IH, IHR, 2*1HT, 1H2, 4*0, 1HF, 1HO, 1HR, 2*1HO, 1H3, 4*0/
Data JL, JP, KL, KP, LL, IP, IC/2*0, 2*0, 2*480, 2*4624, 2*1, 2*0/
Data INP, LL, LP/2*4624, 2*0, 2*0/
Data NMSG/O/
Data MSGO, MSGM, MSGN, MSGC, MSGC/20, 1, 20*0, 20*1, 40*1, 120*0/
Data TEXT/250*1H, 250*1H, 250*1H, 250*1H /

---
Data MIL, NLL, MPH, MLP/20*1, 20*0, 20*1, 20*0/
Data LBLG/1HT, 1HR, 1H, 1HR/

111-134
Annotation and Image Assembly: TRANSAPS 7-7

C ??? Output Image Specifications

C

100 CONTINUE
LIT=1NR
IF(IPRNT.NE.0) LIT=1HP
TYPE 510,LIT
510 FORMAT(//,3X,"OUTPUT IMAGE SPECIFICATIONS: ",/5X,
* "OUTPUT FILE NODE: ",/6X,
* "R = GRAY SCALE RASTER IMAGE FILE "ANNING.DAT", "/8X,
* "P = LINE PRINTER PSEUDO IMAGE FILE "APRINT.DAT", "/8X,
* "5X,"NODE? (R P) ",AI)
ACCEPT 1,LIT
IF((LIT.EQ.1MR),OR(LIT.EQ.1MP)) IPRINT=0
IF((LIT.EQ.1MH),OR(LIT.EQ.1HP)) IPRINT=1
TYPE 520,NL
520 FORMAT(//,5X,"NUMBER OF LINES?",15,5X,"(/ = NO CHNG")
ACCEPT 9,NL
IF(NP.GT.NPIX) NPIX=128
IF(NP.GT.NPIX) NPIX=NP

C ??? Embedded Image Specifications

C

530 FORMAT(5X,"COMPLEMENT BACKGROUND? (Y OR N) ",AI)
ACCEPT 1,LIT
IF((LIT.EQ.1MR),OR(LIT.EQ.1HP)) KOMP=0
IF((LIT.EQ.1MH),OR(LIT.EQ.1HP)) KOMP=1

C

540 FORMAT(5X,"NUMBER OF IMAGES? (0 1 2)",15,5X,"(/ = NO CHNG")
ACCEPT 0,NIMG
IF(NIMG.LT.0) NIMG=0
IF(NIMG.GT.2) NIMG=2
IF(NIMG.EQ.0) GO TO 200
DO 150 I=1,NIMG
150 TYPE 550,1,(FILENAME(J,1),J=1,10)
550 FORMAT(5X,"IMAGE",12,"/",7X,
* "FILENAME (UP TO 9 CHARACTERS) ",10A1)
ACCEPT 1,NAME1
FORMA(T501A1)
IF(NAME1.EQ.IH ) GO TO 140
IF(NAME1.EQ.IH) GO TO 140
IF(NAME1.LE.1H) AND (NAME1.LE.1H) GO TO 120

III-135
TYPE 560
FORMAT(1X,"*** FILENAME MUST START WITH LETTER")
GO TO 110
120 DO 130 J=1,10
FILNAME(J,1)=NAME(LT(J))
IF(NAME(J,1).LE.10) FILNAME(J,1)=0
IF(NAME(J,1).EQ.10) FILNAME(J,1)=0
IF(J.EQ.1) GO TO 130
IF(FILNAME(J-1,1).EQ.0) FILNAME(J,1)=0
130 CONTINUE
140 CONTINUE
TYPE 570, JL(1)
580 FORMAT(7X,"SKIP LINES INTO INPUT IMAGE",15,5X,"(/ = NO CHNG")
ACCEPT 9, JL(1)
TYPE 590, ILUX, KL(1)
580 FORMAT(7X,"SKIP PIXELS INTO INPUT IMAGE (<",16,"">",15,5X,
"(/ = NO CHNG")
ACCEPT 9, ILUX
IF(IPX(1).GE.IPX) IPX(IPX)=0
IF(KL(1).GT.KL) KL(KL)=0
590 FORMAT(7X,"NUMBER OF LINES? (UP TO ",16,"">",15,5X,"(/ = NO CHNG")
ACCEPT 9, KL(1)
IF(KL(1).GT.KL) KL(KL)=0
MAPP=MIP(IPX)-IPX(1)
IF(MAPP,LT,MAXPIX) MAXPIX=MAPP
IF(KP(1).GT.MAXPIX) KP(KP)=MAXPIX
TYPE 600, MAXPIX, KP(1)
600 FORMAT(7X,"NUMBER OF PIXELS? (UP TO ",16,"">",15,5X,"(/ = NO CHNG")
ACCEPT 9, KP(1)
IF(KP(1).GT.MAXPIX) KP(KP)=MAXPIX
INP(IPX(IPX)+KP(1)) ILUX=KL(1)+KL(1)
TYPE 610, ILUX, IL(1)
610 FORMAT(7X,"STARTING LINE? (RANGE 1 ",16,"">",15,5X,
"(/ = NO CHNG")
ACCEPT 9, IL(1)
IF(IL(1).LT.1) IL(1)=1
IF(IL(1).GT.ILUX) IL(1)=ILUX
LL(1)=IL(1)+KL(1)-1
IPUMA=1+KP(1)
TYPE 620, IPU, IP(1)
620 FORMAT(7X,"STARTING PIXEL? (RANGE 1 ",16,"">",16,4X,
"(/ = NO CHNG")
ACCEPT 9, IP(1)
IF(IP(1).LT.1) IP(1)=1
IF(IP(1).GT.IPUMA) IP(1)=IPUMA
LP(1)=IP(1)+KP(1)-1
LIT=INT
IF(IC(1),NE.0) LIT=1NY
TYPE 630, LIT
630 FORMAT(7X,"COMPLEMENT IMAGE? (Y OR N) ",4X)
ACCEPT 1, LIT
IF((LIT.EQ.1NY).OR,(LIT.EQ.1NN)) IC(1)=0
IF((LIT.EQ.1NY).OR,(LIT.EQ.1NN)) IC(1)=1
150 CONTINUE
Embedded Annotation Specifications

TYPE 040, MSG

ACCEPT 0, MSG

IF(MSG.LT.0) MSG=0

IF(MSG.EQ.20) MSG=20

IF(MSG.EQ.0) GO TO 300

DO 320 MSG=1, MSG

IF(MSG.EQ.0) GO TO 310

TYPE 050

FORMAT('//33, 'MESSAGE TO CHANGE? (1 - 12)'.5X, '(/ = NO FURTHER CHNG)'

ACCEPT 0, R

IF(R.LT.1) OR.(R.GT.MSG)) GO TO 300

CALL MSG(R)

GO TO 320

CONTINUE
9F

So

1m*

It's me I see

9601601.,

It's ps

TYPE 750

750  FORMAT(//, 'USER SPECIFICATION COMPLETE: ', //, D8.3, //, //, //,

"RESEND (1 ON 0? 0")

ACCEPT 1,1,1

IF(1,1,1) GO TO 100

IF(1,1,1) GO TO 100

RETURN

END
~~ (

440

eISS4a

hees
ee.*sesl

1v04
I

Siss

.
4I 4104

mee.
IM44440
Se. s

IOt"

cft"#re

~evsesm ese

,.

*gnum

am

Oem

*8

s

'W16406a"I
4o*

641

4Ce

6es

Ao. 101000W M*61 .e"e e*9

4)1 *WS 104

WW

Of * C 1.#*.**
AWPI 3.181m

"%mot~m

3S1**


EXECUTING SYMBOL

PURPOSE: INPUT SYMBOL FILE-SEE LINES FROM FILE SYMBOL.DAT

CALLED FROM ANNOT

>Error (

Cngm /SYMBOL/ 
DIMENSION JSTN(5040)
DIMENSION JSTN(17280)
DIMENSION JJSTN(17280)
DI8E JJSTN
EQUIVALENCE (JJSTN,JSTN)
OPEN(HTR),TYPE="OLD",NAME="SYMBOL",FORM="UNFORMATTED",
REASON)
J=0
JL=1000
DD 110 J=1,0
READ (11)(JSTN(J),JJSTN,JL)
J(J)JL=1
JL(J)=1000
CONTINUE
END
ANNEXE A. ASSEMBLY \(01 \)1
PROGRAMS 7-17

CREATING FILESET

C
C Purpose: open and position image files
C
C Called from "IMAGE"
C

COMMD SENDING FLING,FILEA(10,1),FL(2),LP(2),AP(2),
C(1),LI(1),LL(1),IJ(1),IC(1),
DATA FILEA
C
DIMENSION F(110),F2(10)
DATA FL,F2
END

IF(F(IEM,MLE,3) OPEN(INPUT2,TYPE="ILE",NAME=FILE,FORM="UNFORMATTED",
READ=FL)
IF(F(IEM,MLE,3) OPEN(OUTPUT2,TYPE="ILE",NAME=FILE,FORM="UNFORMATTED",
READ=FL)
IF(F(IEM,MLE,5) GO TO 100
DO 100 IO, IO, IO
100 F(IEM,MLE,5) GO TO 100
CONTINUE
100 CONTINUE
IF(F(FLING,MLE,3) GO TO 100
OPEN(INPUT0,TYPE="ILE",NAME="ABING",FORM="UNFORMATTED")
GO TO 100
100 OPEN(OUTPUT0,TYPE="ILE",NAME="APRINT")
100 RETURN
END

111-145
SUBROUTINE TITSET(L)

C Purpose: Set annotation line segments for all messages active on line L

C Loop on messages

C Branch on orientation (T,B vs L,R)

C Identify Symbol Map lines or columns to extract

C Extract Symbol Map lines or columns

C Replicate lines or columns to required symbol size

C Transfer message segment to output line buffer

C Called from: ANNOT

C Calls: INDEX4, LINES4, CLNBS4, BSPL1, BSPL2, BSPL3, BSPL4

C-----------------------------------------------

COMMON /ENBASEG/, MBG=.BSG(20), ABG(20), ABGC(20), TEXT(50, 20)
* MBG(20), MBGP(20), ABGC(20), TEXT(50, 20)
BITE TEST
COMMON /BOUND/, MLL(20), MLL(20), MLLP(20), MLFP(20)
COMMON /OUTLIN/, LINE(4000)
BITL LINE
COMMON /IMBASEG/, LTMP(4000)
BITE LTMP
COMMON /CLASSG/, BT, AL, AL4, AL4(2), KT, L4T(3, 2), CHAR(64)
BITE CHAK
IF(MBG.EQ.0) GO TO 310
DO 500 M=1, MBG
NTMBG(0)
IF(LT.L4.0) GO TO 300
LFLM=MLL(0)
IF(LT.LT.0) GO TO 500
LFLM=MLL(0)-L
IF(LT.LT.0) GO TO 500
ITM=MBG(0)
NTMBG(0)
L=MBG(0)
J=MF(0)
JL=MLP(0)
DO 710 J=JL
LTM(0)J=0
IF(IT.EQ.0) GO TO 300
AL=LFL1+1
IF(IT.EQ.7) M=ALFL1+1
CALL INDEX4
DO 200 M=1, NT
DO 210 J=1, MX
CHAR(J)=0
200 IF(MB.0) GO TO 220
RT=TEXT(R, M)
CALL LINES4
IF(MT.EQ.1) CALL BSPL1
IF(MT.EQ.2) CALL BSPL2
IF(MT.EQ.3) CALL BSPL3
IF(MT.EQ.4) CALL BSPL4

111-146
220  \( \text{KX} = (K - 1) \times MX \)
IF(1T.EQ.2) GO TO 230
IF(JI.NE.KX)
INC=1
GO TO 240
230  IPT=JI\( \times MX \)
INC=1
240  \( \text{DD} \ 250 \ J=1, MX \)
L1(TP(IPT))=CHAR(J)
250  IPT=IPT+INC
260  CONTINUE
GO TO 400
300  CONTINUE
MC=LFI
IF(1T.EQ.4) MC=LFI
K=MC/\( MX \)
NL=MC+1=K*\( MX \)
K=K+1
CALL INDEX4
DO 310 J=1, MX
CHAR(J)=0
IF(NL4.EQ.0) GO TO 320
K=TEXT(K, \( M \))
CALL CM454
IF(1T.EQ.1) CALL RMSPL1
IF(1T.EQ.2) CALL RMSPL2
IF(1T.EQ.3) CALL RMSPL3
IF(1T.EQ.4) CALL RMSPL4
320  IF(1T.EQ.4) GO TO 330
IPT=JI
INC=1
GO TO 340
330  IPT=JI
INC=1
340  \( \text{DD} \ 350 \ J=1, MX \)
L1(TP(IPT))=CHAR(J)
350  IPT=IPT+INC
400  CONTINUE
IF(\( M \& G C ( \# ) \).EQ.0) GO TO 420
\( \text{DD} \ 410 \ J=1, JLI \)
410  \( \text{LTP}(J) = \text{NOT. L1(TP}(J) \)
420  \( \text{DD} \ 430 \ J=1, JLI \)
430  \( \text{LINE}(J) = L1(TP}(J) \)
500  CONTINUE
510  RETURN

END
SUBROUTINE INDEX4

Purpose: Determine Symbol Map lines or columns to be extracted

N4 = Number of lines/columns for given
    message line, symbol size

L4 = Index of first line/column to be extracted
    from Symbol Map

CALLED from: TXTSET

COMMON /LCLMSG/ MT, ML, N4, NL4, NL4(2), KT, L4T(3), CHAR(64)
BYTE CHAR
DIMENSION N4(64,4), L4(64,4)
BYTE N4,L4
DATA N4/2*0,12*2,2*0,48*0, 4*0,24*2,4*0,32*0,
+ 1,2,1,1,2,1,1,1,2,1,1,2,1,1,1,2,1,1,2,1,1,2,1,
+ 1,2,1,1,2,1,1,1,2,1,6*0,16*0, 8*0,48*1,8*0/
DATA L4/0,2,6,10,14,18,22,26,30,34,38,42,46,0,0,48*0,
+ 4*0,1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,
+ 41,43,45,47,49*0,32*0, 6*0,1,2,4,5,6,8,9,10,12,13,14,16,17,18,
+ 20,21,22,24,25,26,28,29,30,32,33,34,36,37,39,40,41,42,
+ 44,45,46,48,6*0,16*0, 8*0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,
+ 15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,
+ 35,36,37,38,39,40,41,42,43,44,45,46,47,48,8*0/
NL64=NL4(MT)
IF(NL64.EQ.0) GO TO 120
KL4=NL4(MT)-1
DO 110 J=1, KL4
110 NL4(J)=NL4+J
120 RETURN
END
SUBROUTINE LINES4

Purpose: Extract line(s) from Symbol Map (3 words/line)

CALLED from: TXTSET

COMMON /LCUMSG/ NT, ML, ML4, ML4(2), KT, L4T(3, 2), CHAR(64)
BYTE CHAR
COMMON /SYMBOL/ ISYM(3, 40, 60)
DO 120 J=1, ML4
   KL4=ML4(J)
   DO 110 JJ = 1, 3
      L4T(JJ, J) = ISYM(JJ, KL4, KT)
   110 CONTINUE
120 CONTINUE
RETURN
END
SUBROUTINE CLN54

C Purposes: Extract column(s) from Symbol Map (3 words/column)
C
C CALLED from: TXTSET
C
COMMON /LCMGSG/ NT, ML, NL4, ML4(2), KT, LTIT(3,2), CHRN(64)
BYTE CHAR
COMMON /SYMBOL/ ISYM(3,48,60)
DIMENSION JLT(48), JKT(48)
BYTE JLT, JKT
DATA JLT/10,1,16,2,16,3/
DATA JKT/-15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0,
+ -15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0/
DO 130 J=1, NL4
JL4=NL4(J)
JJL=JLT(JL4)
JJJS=JKT(JL4)
KL4=0
DO 120 JJ=1, 3
M=0
DO 110 JJP=1, 16
KL4=KL4+1
ITS=ISYM(JJL, KL4, KT)
JTS=JJJS
M=M+ISHFT(ITS, JTS).AND.1
ITS=M
JTS=16-JJP
M=M+ISHFT(ITS, JTS)
110 CONTINUE
L4T(JJ,J)##
120 CONTINUE
130 CONTINUE
END
SUBROUTINE REMPLI

Purpose: Resample 4x Symbol Map to 1x Symbol size

Symbol Map

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
</tr>
</tbody>
</table>

Resample 1x

| I | J | K | L |

Pixel

I = (F.or.K).and.(G.or.J)

CALLED from: TITSSET

COMMON /LCLASG/ NT, NL, NL4, NL4(2), NT, L4T(3,2), CHAR(64)

BYTE CHAR

JCHR=2

DO 120 JJ=1,J

L4T1=L4T(JJ,1)

L4T2=L4T(JJ,2)

ITS=L4T2

JTS=1

L4F=L4T1.0R.IIFHT(ITS,JTS)

ITS=L4T1

L4F=L4F.AND.(IIFHT(ITS,JTS).0R.L4T2)

JS=-14

DO 110 JP=1,4

JCHR=JCHR+1

ITS=L4F

JTS=JS

NB=IIFHT(ITS,JTS).AND.0.1

IF(NB.NE.0) CHAR(JCHR)="377"

JS=JS+4

110 CONTINUE

RETURN

END
SUBROUTINE REMPL2

Purpose: Resample 4x2 Symbol map to 2x symbol size

Symbol Map          Resample Map
A  B  C  D      U  =  0, and E
E  F  G  H      X  =  C, and H
4x4 cell  I  J  K  L  Pixels  YZ  =  I, and H
M  N  O  P

CALLED FROM: TGETSET

COMMON /LCLARG/ NT,HL,HL4,HL4(2),KT,L4T(2,2),CHAR(64)
BYTE CHAK
IF((HL7.AND.="1"),CO,0) GO TO 110
MSK1="104210"
MSK2="21042"
GO TO 120
110    MSK1="21042"
       MSK2="104210"
120    JCHR=0
       DO 140 JJ=1,3
       L4T1=L4T(JJ,1)
       L4T2=L4T(JJ,2)
       ITB=L4T1
       JTS=1
       L4S1=I18HFT(ITS,JTS),AND,L4T2
       ITS=L4T2
       L4S2=L4T1,AND,118HFT(ITS,JTS)
       L4F=(MSK1,AND,L4S1),OR,(MSK2,AND,L4S2)
       JS=-15
       DO 130 JP=1,9
       JCHR=JCHR+1
       ITB=L4S1
       JTB=JS
       HD=118HFT(ITS,JTS),AND,"1"
       IF(RB,HE,0) CHAR(JCHR)=377
       JS=JS+2
130    CONTINUE
140    RETURN
END

111-152
SUBROUTINE SAMPLE

Purpose: Resolve 4x symbol map to 3x symbol size

Symbol Map    
  A B C D    Resolved RST    E F
  I J K L    Pixel LIT    U V
  M N O P

CALLED FROM: TEST

COMMON /LCLASS/ CT,RL,4L4,RL4(I),RT,LRT(J,2),CHAR(64)
BYTE CHAR
DIMENSION JST(12)
DATA JST/"A"-"Z",-12,-11,-10,-9,-8,-7,-6,-5,-4,-3,-2,0/
JCHRD0
DO 100 J=1,12
LRTS=LRT(JJ,1)
IF(JL4,ML1,110)
GO TO 110

110 LRTS=LRT(JJ,1)
LST=LST(JJ,1)
ITBS=ITBS(ITBS,JTS)
JTS=1
120 J=12 J=1,12
JCHRSJCHRS+1
JCHRSJCHRS+1
ITBS=ITBS+1
JTS+JTS
120 118WFT(ITBS,JTS),AND,"1"
IF(JL4,ML1,1)
GO TO 110
CONTINUE
RETURN
END
SUBROUTINE SAMPLE

C Purpose: assemble 64 symbol map to 64 symbol size
DIRECT MAPPING
C CALLED FROM TESTSET

COMMON //LCLASS//.ST.UL,UTL,ML4.TL,TLT(LT1,L4),ULT,UL𝔨,UL4,CH4(64)
BYTE CH4
JCH4=00
DO 120 J=1,3
LAF=LRT(JJ,1)
J80=15
DO 110 JP1=1,16
JCH4=JC=J+1
LT5=45
JTS=J5
SB=11BT7T(ITS,JTS),AND,."
IF(SB,."1.0) CH9(JCH4)="777
J80=J80=1
110 CONTINUE
RETURN
END